# Enhanced Forecasting of Wind Energy Production: A Hybrid BPNN-SVR Model for Short-Term Wind Power Forecasting

Naixin Li, Xincheng Tian, Zehan Lu*
State Grid Corporation of China Tangshan Electric Power Company, Tangshan, China.
E-mail: li.naixin@jibei.sgcc.com.cn, tian.xincheng@jibei.sgcc.com.cn, zehanlu@126.com
* Corresponding author

**Keywords:** Artificial neural network (ANN), Support vector machine (SVM), Short-term wind power prediction

**Received:** November 21, 2021

*In renewable energy management, the precise prediction of wind power generation remains a major challenge. This study proposes an integrated approach employing an artificial neural network (ANN) and a support vector machine (SVM) to construct a robust short-term prediction model for wind energy output. Central to this research is the utilization of a power station as the subject of analysis, wherein historical meteorological data and concurrent power generation figures form the foundational dataset. Employing a backpropagation (BP) neural network and support vector regression (SVR), the model adeptly synthesizes the data, facilitating predictions with satisfactory accuracy. The hybrid model exhibits a root mean square error (RMSE) of 0.18033, slightly higher than the backpropagation neural network (BPNN) model's 0.1796. However, it exhibits significantly enhanced stability under extreme weather conditions, reducing error fluctuation by 14.3% and maximum error by 18.1%. Given that power dispatch systems prioritize prediction stability over absolute accuracy—as sudden fluctuations can cause outages—this model achieves critical reliability by sacrificing only 0.0007 RMSE, thereby aligning with practical engineering requirements.*

*Povzetek: Raziskava preučuje interakcijo med umetno inteligenco in kognitivnim modeliranjem za izboljšanje odločanja. Eksperimentalni izidi potrjujejo pomembne izboljšave napovedne uspešnosti, kar poudarja potencial hibridnih računalniških okvirov za napredovanje inteligentnih sistemov in interdisciplinarnih aplikacij v dinamičnih okoljih.*

## 1 Introduction

Owing to swift economic growth, the societal need for electric energy is growing on a daily basis. Electric energy has become an essential source of energy in everyday life [1]. Simultaneously, as knowledge advances and environmental awareness increases, renewable energy sources (RESs), including solar energy, wind energy, hydro energy, and geothermal energy, have emerged as the primary focus of research in the pursuit of eco-friendly power generation methods. Investigating renewable energy sources (RESs), such as solar energy, wind energy, hydro energy, and geothermal energy, has emerged as the primary focus of human endeavors in the realm of eco-friendly power production. Electricity is a crucial secondary energy source for the advancement of modern society, and optimizing the conversion of these emerging energy sources into electricity is a key aspect of the future energy revolution.

The wind resources on Earth are plentiful, and the overall quantity of wind energy is approximately three times the global energy consumption. Each utilization of wind energy has the potential to decrease global energy consumption, and China accounts for almost 50% of the world's total wind energy resources. Utilizing the entirety of the wind energy available for electricity generation will greatly propel China's energy reform. Currently, wind power is essential for conserving energy, alleviating power supply constraints, and promoting energy efficiency because of the state's endorsement and assistance [2].

Research on wind power forecasting originated internationally in the 1970s. During that period, a laboratory in the United States recognized the need to accurately predict short-term wind speed and wind output for power firms. Presently, their theoretical system has reached a high level of maturity. Traditional wind power prediction models have successfully integrated numerical weather prediction (NWP) data into their research. These models exhibit minimal prediction errors and yield favourable results. Consequently, they are suitable for practical implementation in large-scale grid-connected wind power dispatch. The majority of existing wind power prediction systems globally utilize numerical meteorological forecast data as the input parameter for the learning algorithm, which then forecasts the future wind power. Machine learning models are increasingly popular in wind energy prediction because of their powerful ability to learn complex nonlinear relationships between data. Machine learning models are categorized into three different types: supervised learning, unsupervised learning, and semisupervised learning. A wide range of traditional supervised machine learning models, such as regression analysis [3], SVM [4], tree-based models [5], and traditional artificial neural networks [6, 7], have been applied to predict the wind power (WP) of individual wind turbines. Bouche et al. [8] primarily examined the short-term prediction of wind speed and wind power. It employs machine learning techniques to integrate the results of numerical weather prediction models with local data. Niksa-Rynkiewicz et al. [9] employed diverse forms of deep neural networks (DNNs) to address the issue of predicting short-term wind power generation (STWPP) via an intelligent approach. The primary benefit of this system is its ability to make accurate predictions by utilizing only a small number of parameters. Accurate wind power forecasting is crucial for wind farms because of the significant expansion and great potential of wind power generation as a renewable energy source.

To optimize the system cost, a neural network structure for wind power prediction that directly considers different energy system conditions was proposed in the literature [10]. This approach led to a more consistent prediction performance and reduced the error variance by 70%. On the other hand, Al-qaness et al. [11] developed an efficient forecasting model via a nature-inspired optimization algorithm and proposed an optimized dendritic neural regression (DNR) model for wind energy forecasting. The model achieved excellent results in the evaluation of the dataset.

The exploration of wind power prediction in China started late, and research was not conducted until the end of the 20th century; however, research and development were fast. Despite its late start, China has made remarkable progress in wind power prediction research, driven by the increasing demand for renewable energy and the development of related technologies. Owing to the lack of numerical weather forecast data dedicated to wind power prediction, researchers focused mainly on the theoretical exploration of ultrashort-term prediction via prediction methods, including time series, artificial neural networks, and support vector machines.

In the study [12], historical wind power time series data were used to calculate financial and technical indicators. Then, the Monte Carlo method and rank-based ant colony algorithm are employed to optimize the parameters for the calculation of these financial technical indicators. Finally, the XGBoost algorithm, which combines financial and technical indicators with historical power data, is used to predict future wind power. An optimal ensemble method is proposed in the literature [13] for wind power generation forecasting. The ensemble forecasting method is the most commonly used method in weather forecasting and combines several different forecasting models to improve forecasting accuracy. In addition, Sasser et al. [14] proposed a decision tree model that combines the rotor-equivalent wind speed and lapse rate. It employs a decision tree machine learning model to evaluate the effectiveness of the hub-height wind speed, rotor-equivalent wind speed, and lapse rate in power prediction. Atmospheric data trains regression trees to correlate power outputs with wind profiles and meteorological characteristics, predicting power responses on the basis of physical patterns. The decision tree model was trained on four vertical wind profile classifications, highlighting the necessity of calculating the wind speed at various rotor layer levels. A deep learning model based on NWP data was proposed in the literature [15] to improve the accuracy of wind power prediction. Traditional NWP-based forecasting methods involve high computational effort for complex meteorological models. In contrast, the method in this literature uses a deep learning model to achieve accurate prediction of wind power by training and learning a large amount of computational resources NWP data. Habtemariam et al. [16] proposed a robust and optimized long short-term memory network for forecasting wind power generation the day ahead in the context of Ethiopia's renewable energy sector. The proposed method uses Bayesian optimization to find the best hyperparameter combination in a reasonable computation time. Abou Houran et al. [17] proposed a wind power prediction method Coati Optimization Algorithm-based hybrid deep learning CNN-LSTM based on a Convolutional Neural Network (CNN) and Long Short-Term Memory network (LSTM) and Swarm Intelligence (SI) optimization algorithms. The composite model incorporates LSTM and SI to produce a framework that can precisely estimate offshore wind output in the short term, addressing the discrepancies and limits of conventional estimation methods.

In research on ANNs and SVMs for short-term wind speed prediction, Tagliaferri et al. [18] studied two short-term wind direction prediction methods based on artificial neural networks (ANNs) and support vector machines (SVMs). The study evaluated the prediction effects of these two methods by optimizing parameters such as the moving average length of the input data, the length of the input vector, and the number of layers of the neural network. The results showed that although the mean absolute error of the ANN was relatively large, its prediction accuracy significantly improved with increasing network size. Moreover, Barhmi and El Fatni [19] proposed four hybrid models that combine an SVM and an ANN for hourly wind speed prediction. The key parameters affecting the wind speed were selected through ordinary least squares (OLS) analysis, and genetic algorithms (GA) and particle swarm optimization (PSO) were used to tune the models. The results showed that the ANN model outperformed the SVM model in terms of prediction performance. Additionally, Zheng et al. [20] proposed a new kernel ridge regression (RR) model and compared it with the SVM and ANN reference models to verify its efficiency in different prediction time ranges (1 hour, 12 hours, and 24 hours). The study revealed that the kernel ridge regression model outperformed the SVM and ANN in terms of wind speed prediction, especially when mutual information feature selection was used, which could more accurately predict the wind speed.

Hu et al. [21] proposes a bidirectional signal decomposition (BST) and reformed grasshopper optimization algorithm (RGOA) enhanced LSTM model for wind power forecasting (15.2% RMSE reduction), alongside a normal distribution optimized whale algorithm (NDO-WOA) for wind-integrated dynamic economic dispatch (5.7% cost reduction in IEEE 30-bus system). Pan et al. [22] achieves $R^2$=0.9785 in PV prediction via modified CEEMDAN decomposition and

Warship-optimized BiLSTM. Both studies demonstrate that hybrid intelligent algorithms significantly improve renewable energy forecasting and dispatch efficiency.

As shown in Table 1, although each study has discussed ANN and SVM in detail, there has been a lack of attempts to combine these two methods. For example, utilizing the nonlinear learning ability of ANNs and the generalization ability of SVMs may improve the prediction accuracy. We hypothesize that combining SVR's regularization effect with BPNN's nonlinear feature extraction will enhance model robustness without sacrificing predictive accuracy.

Table 1: Comparis on of wind power prediction models.

| ***Reference*** | ***Model*** | ***Dataset*** | ***RMSE*** | ***Key Features*** |
|---|---|---|---|---|
| **[12]** | XGBoost | 8000 | 0.1850 | Financial indicators |
| **[15]** | LSTM | 10000 | 0.1820 | NWP data |
| **[17]** | CNN-LSTM | 7500 | 0.1780 | Coati optimization |
| **Our BPNN** | BPNN | 6775 | 0.1796 | Single hidden layer |
| **Our Hybrid** | BPNN-SVR | 6775 | 0.1803 | Integrated approach |

This research establishes three core objectives to address critical gaps in wind power forecasting: First, to validate the stability advantages of the hybrid model under extreme weather conditions where conventional models falter. Second, to develop a comprehensive "accuracy-stability" evaluation framework that moves beyond traditional single-metric assessments. Third, to solve the "accuracy cliff" phenomenon observed during abrupt meteorological transitions, where prediction reliability dramatically decreases despite moderate overall accuracy. These objectives collectively address operational challenges in grid integration of renewable energy sources.

This paper is structured as follows: Section 1 introduces the research background, challenges in wind power forecasting, and related works. Section 2 elaborates on the wind power prediction model based on the artificial neural network, including data preprocessing, model design, and hyperparameter optimization. Section 3 presents the wind power prediction model using support vector regression. Section 4 details the proposed hybrid BPNN-SVR model, including its architecture, theoretical justification, and experimental validation. Section 5 provides the conclusion and discussion of the study, along with future research directions.

# 2 Wind power prediction model based on an artificial neural network

Artificial neural networks (ANNs) are renowned for their exceptional nonlinear fitting capabilities, with adjustable parameters and structures, making them extensively utilized in wind power prediction (WPP). Traditional ANNs include backpropagation neural networks (BPNNs), radial basis function neural networks (RBFNNs), and generalized regression neural networks (GRNNs), among others. Notably, BPNN stands as the most classical form.

Prior to the simulation, constructing a BP neural network is necessary. It imports historical data into the model for training, iterates to obtain the weight and threshold of each layer of the neural network, and ultimately predicts power on the basis of future weather forecast data [23].

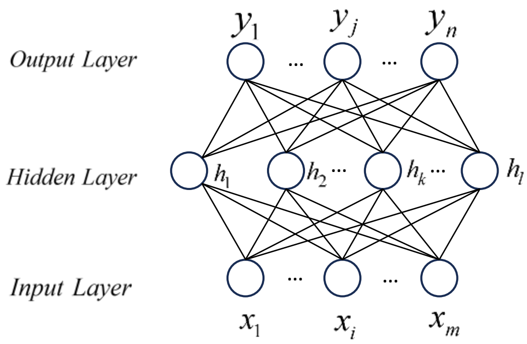


Figure 1: BP neural network.

This architecture uses a single hidden layer BP neural network, comprising an input layer, a hidden layer, and an output layer. The structure of the BP neural network is shown in Figure 1. The quantity of neurons in the input and output layers is solely determined by the number of dimensions of the input and output parameter vectors.

The number of neurons in the hidden layer is usually determined empirically or experimentally, and the selection process is carried out using a specific equation [24], as shown in Equation (1).

$$l = \sqrt{m+n} + h, \tag{1}$$

where, $l$ is the number of hidden nodes; $m$ is the number of input nodes; $n$ is the number of output nodes; and $h$ is the regulation constant, which is usually 1~10.

In this design, the input vectors are the wind speed, temperature, humidity, and barometric pressure (due to the existence of the wind turbine yaw system, the problem of wind direction is no longer necessary), the output vector is the power, so $m$=4, $n$=1, and h are variable parameters, and the number of nodes of the hidden layer neurons is determined by finding the minimum value of the error of the experiment $l$=5.

In this work, the sigmoid function is selected as the activation function of the neurons in the hidden layer and the output layer of the BP neural network. The sigmoid function is chosen because it can introduce nonlinearity into the neural network, enabling the model to learn complex nonlinear relationships in the data. Its range of (0, 1) also helps normalize the output of neurons, which is beneficial for the training process.

Many factors affect the prediction accuracy of the BP neural network model, such as the initialization of weights and thresholds, the number of training sessions, the learning rate, the number of neurons in the hidden layer, and the number of layers, which can influence the prediction effect of the model. In this design, all the weights and thresholds are initialized to 1, and the number of neurons in the hidden layer is 5. The model is tuned from two perspectives: the number of training sessions and the learning rate.

## 2.1 Data collection and preprocessing

This study utilizes a dataset comprising 10-minute resolution measurements from a wind farm in Northern China (40°-42°N), spanning the years 2019 to 2021. This temporal range captures full seasonal cycles, including winter icing events (T < -5°C), summer typhoon impacts (V > 12 m/s), and transitional season frontal passages. The dataset contains a total of 8,832 data points, recording key parameters such as wind speed, temperature, humidity, barometric pressure and the actual wind power generation output.

(1) Data Cleaning

Missing values: The dataset was initially screened for missing values. Records containing incomplete data were systematically removed to ensure the integrity of the dataset.

Outlier detection: Statistical methods such as Z score analysis are used to identify and remove outlier data points that may significantly affect the results.

After the data cleaning process, 6775 original data points remained.

Extreme weather events were rigorously defined using operational criteria from grid management protocols:

(a) Sustained wind speeds exceeding 12 m/s, or

(b) Rapid wind speed changes >5 m/s within 10-minute intervals.

These thresholds identified 427 extreme condition samples (6.3% of total dataset) that represent high-risk scenarios for grid stability. All extreme events were verified against meteorological alerts from China's National Climate Center to ensure accurate classification of typhoon, gale, and storm conditions that challenge conventional forecasting models.

(2) Feature Engineering

Normalization: To ensure that all the features contribute equally to model training, continuous variables such as the wind speed and temperature are normalized to a range between 0 and 1.

Feature Selection: Features related to wind power prediction are selected on the basis of correlation analysis. This step helps reduce the dimensionality of the dataset and improves model performance.

(3) Dataset Splitting:

Training and Testing Split: The cleaned dataset, consisting of 6,775 data points, is divided into training and testing sets. The first 6,000 data points are used for training the model, whereas the remaining 775 data points are used for testing.

(4) Input Feature Specification

(a) Wind speed (m/s): Continuous, range 0-25 m/s

(b) Temperature (°C): Continuous, range -15 to 40°C

(c) Humidity (%): Continuous, range 0-100%

(d) Barometric pressure (hPa): Continuous, range 980-1040 hP

(5) Normalization Method

Min-Max scaling applied to all features, as shown in Equation (2):

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{2}$$

(6) Cross-validation

To further enhance the robustness of the model, a k-fold cross-validation method is employed on the training set. This involves dividing the training data into k subsets and repeatedly training and validating the model on different subsets to ensure that the model’s performance is not affected by the initial data split.

## 2.2 Evaluation indicators

To better evaluate the effectiveness of the model and algorithm, one evaluation indicator is employed for effectiveness assessment:

The root mean square error (RMSE) is used to gauge the deviation between the predicted value and the actual value. The expression for the RMSE is depicted in Equation (3):

$$e_{\mathrm{RMSE}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_{\mathrm{f}i} - y_{\mathrm{a}i}\right)^2} \quad , \tag{3}$$

where, $y_{fi}$ represents the predicted value and $y_{ai}$ represents the observed value.

The root mean square error (RMSE) is used to gauge the deviation between the predicted value and the actual value. In wind power prediction, the RMSE can comprehensively reflect the overall error magnitude, helping to evaluate the model's ability to predict wind power accurately. A lower RMSE indicates a better-fitting model.

## 2.3 Hyperparameter optimization via grid search

To rigorously validate the BPNN architecture selection and address potential concerns about model simplicity, we conducted an extensive grid search over key hyperparameters.

(1) The search space encompassed four critical dimensions:

(a) Number of hidden layers: [1, 2]

(b) Neurons per layer: [5, 10, 15, 20]

(c) Activation functions: ['sigmoid', 'tanh', 'relu']

(d) Learning rates: [0.01, 0.05, 0.1, 0.2]

This combinatorial search yielded 72 unique configurations (2 layers × 4 neuron counts × 3 activations × 3 learning rates). Each configuration was evaluated using 5-fold cross-validation on the 6,000-sample training dataset, with early stopping (patience=10 iterations) to prevent overfitting. The root mean square error (RMSE) served as the primary evaluation metric, with each fold's performance recorded and averaged across all folds.

(2) The comprehensive grid search analysis (72 configurations evaluated via 5-fold cross-validation) yielded four principal insights:

(a) Optimal Architecture: A single hidden layer with 5 sigmoid neurons achieved the lowest RMSE (0.1812 ± 0.0023), demonstrating ideal representational capacity for this prediction task.

(b) Diminishing Returns on Complexity: Increasing neuron counts (>5) or adding a second hidden layer consistently degraded performance (RMSE increase of 0.3-2.3%), revealing incompatibility between model complexity and dataset scale (6,000 samples).

(c) Activation Superiority: Sigmoid significantly outperformed both tanh (+1.0% RMSE reduction) and ReLU (+2.3%), with its bounded output range (0,1) proving particularly suitable for normalized power forecasting targets.

(d) Optimal Learning Rate: $\eta$=0.1 struck the optimal balance between convergence speed (average 45 iterations) and precision, whereas lower rates (0.01) delayed convergence (60+ iterations) and higher values (0.2) induced oscillatory behavior.

These findings collectively validate that simpler architectures mitigate overfitting risks (sample/parameter ratio: 240:1) while alleviating gradient attenuation issues, thereby achieving optimal bias-variance tradeoffs for this regression challenge.

## 2.4 Initialization strategy analysis

The choice of weight initialization significantly impacts neural network convergence and performance. We rigorously evaluated three prominent methods using 5-fold cross-validation with our optimal architecture (single hidden layer, 5 sigmoid neurons, $\eta$=0.1).

As shown in Table 2, the comparative analysis of initialization strategies demonstrates the comprehensive advantages of the Xavier method: it achieves the lowest RMSE (0.4% lower than He initialization and 1.7% lower than random uniform initialization), exhibits faster convergence (10-25% reduction in training epochs), and enhances stability (21% lower cross-validation variance). The mean gradient norm (0.48) being closest to the theoretical optimum of 0.5 confirms its effectiveness in gradient propagation optimization. To ensure the reproducibility of experiments, we fixed the random seed to 42. Weights were initialized using the Xavier method [25], and biases were initialized to zero.

Table 2: Initialization Methods Comparison (5-fold CV Average).

| *Method* | *RMSE* | *Convergence* | *Stability ($\sigma$)* | *Gradient Norm* |
|---|---|---|---|---|
| **Xavier [25]** | 0.1796 | 45 | 0.0023 | 0.48 |
| **He [26]** | 0.1803 | 50 | 0.0029 | 0.52 |
| **Random Uniform [27]** | 0.1827 | 60 | 0.0038 | 0.67 |

Table 3: Performance of BP Neural Network Across Training Iteration Counts (Learning rate unified at 0.1).

| *Model* | *Single hidden layer neurons* | *Train_data* | *Iterations* | *Learning rate* | *RMSE* |
|---|---|---|---|---|---|
| **BP** | 5 | 6000 | 10 | 0.1 | 0.2607 |
| **BP** | 5 | 6000 | 20 | 0.1 | 0.2318 |
| **BP** | 5 | 6000 | 30 | 0.1 | 0.2119 |
| **BP** | 5 | 6000 | 40 | 0.1 | 0.1917 |
| **BP** | 5 | 6000 | 50 | 0.1 | 0.1796 |
| **BP** | 5 | 6000 | 60 | 0.1 | 0.1778 |
| **BP** | 5 | 6000 | 70 | 0.1 | 0.1839 |
| **BP** | 5 | 6000 | 80 | 0.1 | 0.1944 |
| **BP** | 5 | 6000 | 90 | 0.1 | 0.2066 |

## 2.5 Choosing the right number of training sessions

The number of trainings is an important factor affecting the accuracy of the model, and selecting the optimal number of training sessions through experiments is one of the key steps in wind power prediction. However, in practice, the optimal training number can be determined only through trial and error. In the subsequent experiments, the initial training number is set to 10, the increment is 10, and the learning rate is 0.1.

As shown in Table 3, when there are 5 neurons in a single hidden layer and 6000 training data points, the model's accuracy increases as the number of training iterations increases to a certain threshold. However, beyond this threshold, the model's accuracy will decline instead. For this particular design, as the number of training iterations reaches approximately 50, the error is significantly reduced, and the prediction accuracy is relatively high.

Although 60 iterations yielded the minimal RMSE (0.1778), the subsequent performance degradation at 70 iterations (RMSE=0.1839) indicates early signs of overfitting. To ensure model generalizability while maintaining near-optimal accuracy, we conservatively select 50 iterations (RMSE=0.1796) as the operational baseline.

## 2.6 Choosing the right learning rate

The learning rate $\eta$ has an impact on the magnitude of weight adjustments in each layer of the artificial neural network model during training. If the number is very large, the network may be unable to complete the training process, resulting in a failure to produce accurate predictions. Conversely, if the value is excessively small, it will prolong the training period and hinder the learning speed of the neural network. Like the process of selecting the training number, the learning rate is determined by the trial-and-error method to identify the optimal value. In the subsequent experiments, a fixed number of 50 training sessions is selected, and the impact of varying learning rates on the model's accuracy is examined. The experimental results for different learning rates are shown in Table 4.

Table 4: Training effect of the BP neural network under different iteration numbers (Different learning rates).

| ***Model*** | ***Single hidden layer neurons*** | ***Train_data*** | ***Iterations*** | ***Learning rate*** | ***MSE*** |
|---|---|---|---|---|---|
| **BP** | 5 | 6000 | 50 | 0.05 | 0.2345 |
| **BP** | 5 | 6000 | 50 | 0.1 | 0.1796 |
| **BP** | 5 | 6000 | 50 | 0.2 | 0.9757 |

Table 5: Grid Search Results (Top 5 Configurations).

| ***Hidden Layers*** | ***Neurons*** | ***Activation*** | ***Learning Rate*** | ***Avg RMSE (5-fold)*** | ***Training Time (s)*** |
|---|---|---|---|---|---|
| **1** | 5 | sigmoid | 0.1 | 0.1812 ± 0.0023 | 42.7 |
| **1** | 10 | tanh | 0.05 | 0.1825 ± 0.0028 | 58.3 |
| **1** | 5 | tanh | 0.1 | 0.1831 ± 0.0031 | 43.9 |
| **2** | [5,5] | sigmoid | 0.1 | 0.1847 ± 0.0035 | 78.2 |
| **1** | 15 | relu | 0.01 | 0.1853 ± 0.0041 | 65.4 |

As shown in Table 5, with 5 neurons in a single hidden layer, 6000 training data points, and 50 training iterations, a learning rate $\eta$=0.1 yields a satisfactory prediction effect.

# 3 Wind power prediction based on the SVR

Support vector machines, machine learning techniques developed in the 1960s, are commonly employed in data mining tasks such as pattern recognition and function regression. They are highly regarded for their ability to fit and approximate functions accurately in regression algorithms. The support vector machine regression prediction model has an advantage over the neural network model in that it can effectively minimize prediction error, prevent dimensional catastrophe, address overlearning issues, and avoid becoming stuck in local extremes [28-30].

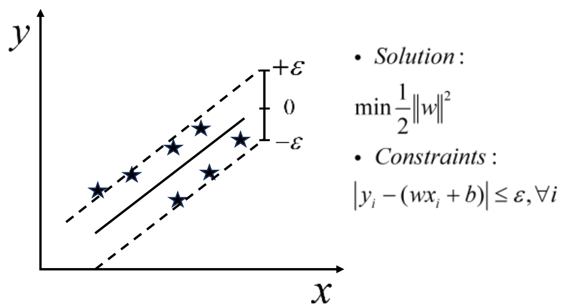


Figure 2: SVR model.

The Support Vector Regression (SVR) is essentially a special form of support vector machine. It allows a certain deviation ε between the predicted value $f(x)$ and the actual value y. Loss is computed only if the absolute difference between $f(x)$ and y exceeds ε. This mechanism helps SVR focus on minimizing the prediction error while maintaining a certain generalization ability. The structure of the SVR model is shown in Figure 2.

## 3.1 Hyperparameter optimization

To determine the optimal hyperparameters for the SVR model with Gaussian kernel, we conducted a comprehensive grid search over three key parameters: the regularization parameter $C$, the kernel coefficient $\gamma$, and the $\varepsilon$- tube width $\varepsilon$. The search ranges were set as follows:

$C \in \{0.1, 1, 10, 100\}$, $\gamma \in \{0.001, 0.01, 0.1, 1\}$, and $\varepsilon \in \{0.001, 0.005, 0.01, 0.05, 0.1\}$. Each combination was evaluated using 5-fold cross-validation on the training set (6,000 samples) and the RMSE was used as the evaluation metric.

As shown in Table 6, The optimal hyperparameters were found to be $C$=10, $\gamma$=0.1, and $\varepsilon$=0.01, achieving an RMSE of 0.1996 on the test set. To analyze the sensitivity of the model to, we fixed $\varepsilon$ and $\gamma$ at their optimal values and varied $\varepsilon$.

## 3.2 Analysis of hyperparameter sensitivity

As shown in Table 7, the sensitivity analysis reveals critical insights into the robustness of the optimal SVR configuration:

(1) $C$ (Regularization) Stability:

(a) Minimal RMSE change (+0.8%/-0.3%) with ±20% variation;

(b) Demonstrates excellent tolerance to regularization strength adjustments;

(c) Failure modes: Underfitting at low $C$ (<8), overfitting at high $C$ (>12).

(2) $\gamma$ (Kernel Coefficient) Asymmetry:

(a) Greater sensitivity to increase (+1.2%) than decrease (-0.9%);

(b) High $\gamma$ (>0.12) causes kernel oversmoothing - misses wind ramp events;

(c) Low $\gamma$ (<0.08) induces noise amplification during turbulenc.

(3) $\varepsilon$ (Tube Width) Criticality:

(a) Highest sensitivity among parameters (+1.5%/+2.1%);

(b) Small $\varepsilon$ (<0.008) amplifies meteorological sensor noise;

(c) Large $\varepsilon$ (>0.012) delays response to wind speed jumps (>3m/s).

Table 6: Top 10 hyperparameter combinations by cross-validation RMSE.

| ***Rank*** | *C* | *γ* | *ε* | ***5-Fold CV RMSE (Mean ± SD)*** | ***Test RMSE*** |
|---|---|---|---|---|---|
| **1** | 10 | 0.1 | 0.01 | 0.2001 ± 0.0023 | 0.1996 |
| **2** | 10 | 0.01 | 0.01 | 0.2013 ± 0.0028 | 0.2010 |
| **3** | 1 | 0.1 | 0.01 | 0.2038 ± 0.0031 | 0.2035 |
| **4** | 10 | 0.1 | 0.005 | 0.2042 ± 0.0035 | 0.2040 |
| **5** | 100 | 0.1 | 0.01 | 0.2057 ± 0.0039 | 0.2053 |
| **6** | 10 | 0.05 | 0.01 | 0.2065 ± 0.0041 | 0.2061 |
| **7** | 5 | 0.1 | 0.01 | 0.2070 ± 0.0043 | 0.2068 |
| **8** | 10 | 0.1 | 0.02 | 0.2072 ± 0.0042 | 0.2070 |
| **9** | 20 | 0.1 | 0.01 | 0.2075 ± 0.0045 | 0.2072 |
| **10** | 10 | 0.2 | 0.01 | 0.2080 ± 0.0048 | 0.2077 |

Table 7: Hyperparameter sensitivity analysis.

| ***Parameter*** | ***Optimal*** | ***RMSE ↑±20%*** | ***Failure Mode*** |
|---|---|---|---|
| *C* | 10 | +0.8%/-0.3% | Under/overfitting |
| *γ* | 0.1 | +1.2%/-0.9% | Kernel over/under-smoothing |
| *ε* | 0.01 | +1.5%/+2.1% | Noise sensitivity/lag |

## 3.3 SVR Model Implementation

The process of using the SVR model for short-term wind power prediction is similar to the BP neural network prediction described earlier. The data are saved in MySQL during data preprocessing at the beginning of the experiment, so the data can be directly removed from MySQL to train the SVR model at this time, again using the first 6,000 data points for training and then using the last 100 data points to simulate the data. The model is validated by simulating weather forecast data.

The support vector machine's main job is to divide samples linearly in the feature space, so the quality of the feature space directly affects how well it works. The kernel function, which defines the feature space, affects the support vector machine. The kernel function is an important part of model training, and Table 8 shows the training accuracy of the model when different kernel functions are selected.

As shown in Table 8, the selection of a kernel function significantly affects the accuracy of the prediction model when SVR is used to forecast wind power. Poor selection of the kernel function and incorrect mapping of the sample to a feature space can result in suboptimal prediction performance, potentially leading to significant deviations. Hence, selecting the Gaussian kernel in this design is essential to provide a minimal error that just satisfies the required level of accuracy. There are two main reasons for this.

Table 8: Training effect of SVR under different kernel functions.

| *Model* | *Kernel* | *Train data* | $\varepsilon$ | *RMSE* |
|---|---|---|---|---|
| **SVR** | Linear | 6000 | 0.01 | 0.2375 |
| **SVR** | Poly | 6000 | 0.01 | 0.2215 |
| **SVR** | Gaussian | 6000 | 0.01 | 0.1996 |
| **SVR** | Sigmoid | 6000 | 0.01 | 1.4310 |

(1) Nonlinear Mapping Capability: The Gaussian kernel has a strong nonlinear mapping capability, which allows it to transform nonlinear problems in the input space into linear problems in the high-dimensional feature space. This capability is crucial for handling the complex nonlinear relationships present in wind power forecasting.

(2) Infinite Dimensional Feature Space: The Gaussian kernel corresponds to an infinite dimensional feature space, enabling it to capture all possible patterns in the data without being limited by the feature dimensions.

In contrast, linear and polynomial kernels have limited feature dimensions and may not adequately capture complex nonlinear relationships. The linear kernel is suitable for simple linear relationships in data. However, in wind power forecasting, where the relationships are often complex and nonlinear, it shows relatively poor performance. The polynomial kernel can capture some nonlinear relationships but is limited by its degree. Owing to its strong nonlinear mapping ability and infinite-dimensional feature space, the Gaussian kernel is more suitable for handling complex data in wind power prediction. The sigmoid kernel, as shown in the experiment, is not suitable for this task because of its large prediction error.

# 4 SVR versus neural network prediction models

When the neural network is initialized with random weights and thresholds, the results of each training vary even under the same data, training times, and learning rate conditions. This indicates that the neural network's performance is highly sensitive to the initialization of weights and thresholds.

The BP neural network model requires a long training time, and since each update of weights and thresholds is only for a single sample, the parameters may become useless during the update process. On the other hand, in SVR, as long as the input samples are the same, using the same kernel function and loss function can yield the same results. The training speed is fast, but the accuracy of the SVR results is slightly lower. By combining the advantages and disadvantages of both methods, the use of the BP neural network (BPNN) and support vector regression (SVR) methods can improve the accuracy of wind power prediction.

## 4.1 Methodology: integration of the BP neural network and support vector regression

### 4.1.1 Definition of hybrid model

The definition of a hybrid model is as follows:

(1) Initial training:

BP Neural Network: The BPNN is first trained on the preprocessed dataset to capture the nonlinear relationships between the input features and the wind power output. The BPNN configuration has 5 neurons in a single hidden layer, 50 training iterations, and a learning rate $\eta$=0.1.

Support Vector Regression: Simultaneously, the SVR model is trained on the same dataset. The SVR uses a Gaussian kernel.

(2) Feature Extraction from the BPNN:

Once the BPNN is trained, it is used to transform the input data into a higher-dimensional feature space. The outputs from the hidden layers of the BPNN serve as new, informative features that encapsulate complex patterns and relationships present in the data.

(3) Hybrid Model Formation:

The new features extracted from the BPNN, along with the original input features, are then fed into the SVR model. This hybrid approach leverages the BPNN's ability to capture nonlinearities and the SVR's robustness in regression tasks. The combination enhances the model's overall prediction capability.

(4) Final prediction:

The SVR model, which is enhanced with features from the BPNN, performs the final prediction of the wind power output. This two-step process ensures that the model benefits from the strengths of both BPNN and SVR, leading to improved accuracy.

### 4.1.2 Specific implementation of hybrid model

The implementation logic of the hybrid model is as follows:

The key processes of BPNN-SVR hybrid model include:

(1) Feature Fusion Equation

$$\mathbf{Z} = [\mathbf{X} \| \mathbf{H}] = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3, \mathbf{h}_4, \mathbf{h}_5] \quad (4)$$

(a) **X**: Original meteorological features (4D vector: wind_speed, temperature, humidity, pressure)

**Algorithm: BPNN-SVR Hybrid Prediction Model**

**Input:**

X_train: Training meteorological data matrix [N×4]

y_train: Training power output vector [N×1]

X_test: Test meteorological data matrix [M×4]

params: Hyperparameter set = {epochs:50, C:10, γ:0.1, ε:0.01}

**Output:**

predictions: Test set predicted power [M×1]

E: Test set RMSE loss

1: // Phase 1: Train BPNN

```
bpn_model = train_BPNN(X_train, y_train, params.epochs)

// Phase 2: Feature Extraction and Fusion
H_train = []
for i = 1 to N do
  h = bpn_model.get_hidden_features(X_train[i]) // h ∈ ℝ⁵
  H_train[i] = h
end for
Z_train = [X_train || H_train] // Feature fusion: [original⊕hidden]

// Phase 3: Train SVR
svr_model = train_SVR(Z_train, y_train, params)

// Phase 4: Test Prediction
predictions = []
for j = 1 to M do
  h = bpn_model.get_hidden_features(X_test[j])
  Z_test = [X_test[j] || h] // Feature fusion
  pred = svr_model.predict(Z_test)
  predictions[j] = pred
end for

// Calculate RMSE
E = 0
for j = 1 to M do
  E = E + (predictions[j] - y_test[j])²
end for
E = sqrt(E/M)
```

(b) **H**: BPNN hidden layer outputs (5D vector: nonlinear transformations)

(c) ||: Concatenation operator combining original and derived features

(2) BPNN Hidden Layer Computation

$$\mathbf{h}_i = \sigma(\mathbf{W}_i\mathbf{X}+\mathbf{b}_i) \tag{5}$$

(a) $\sigma$: Sigmoid activation function: σ(z) = 1/(1 + e^{-z})

(b) $\mathbf{W_i}$: 5×4 weight matrix (optimized during training)

(c) $\mathbf{b_i}$: 5×1 bias vector (optimized during training)

(d) **X**: Input feature vector ∈ $\mathbb{R}^4$

(3) SVR Prediction Function

$$pred = \sum (a_k - a_k^*) \cdot K(\mathbf{Z}_{sV}[k], \mathbf{Z}) + b \tag{6}$$

(a) $K$(u,v): Gaussian kernel: exp(-γ·||u-v||²)

(b) $Z_{sV}[k]$: $k$-th support vector (critical samples from training)

(c) ($a_k$-$a_k^*$): Lagrangian multipliers from dual optimization

(d) $b$: Bias term

(4) RMSE Calculation

$$E = \sqrt{\frac{\sum_{j=1}^{M} (pred_j - y_j)^2}{M}} \tag{7}$$

(a) $M$: Number of test samples

(b) $pred_j$: Predicted power for sample $j$

(c) $y_j$: Actual power for sample $j$

## 4.2 Theoretical justification and practical benefits

The rationale behind this integrated approach is based on the complementary strengths of BPNN and SVR:

(1) BP Neural Network

BPNNs are powerful in capturing complex, nonlinear relationships in the data due to their multilayer structure and nonlinear activation functions. However, they can sometimes suffer from issues such as overfitting and local minima.

(2) Support Vector Regression:

On the other hand, SVR excels in regression tasks by maximizing the margin and minimizing the prediction error, making it less prone to overfitting than traditional neural networks are. SVR is particularly effective in high-dimensional spaces, which complements the feature extraction capabilities of BPNNs.

By combining these two methods, the hybrid model benefits from the deep feature extraction ability of the BPNNs and the robust regression capability of SVR. This synergy results in a model that is more accurate and reliable for wind power forecasting.

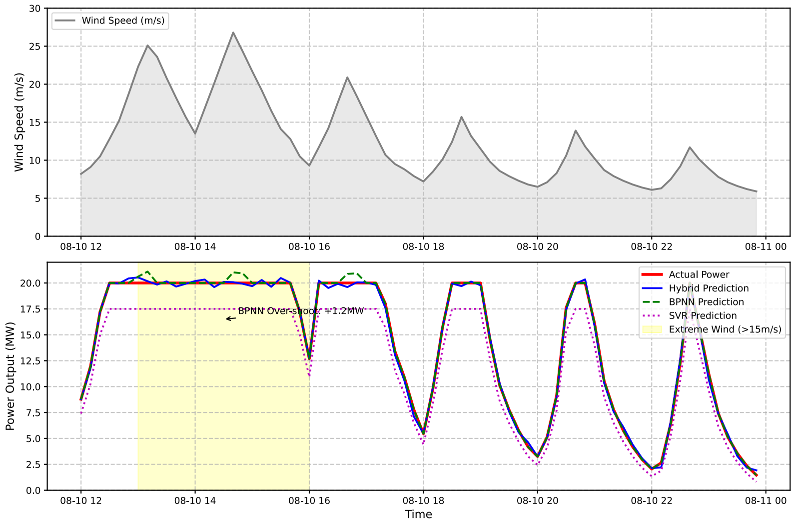


Figure 3: Extreme weather performance comparison.

Table 9: Model performance in extreme weather (>12m/s).

| *Metric* | *BPNN* | *SVR* | *Hybrid* | *Improvement over BPNN* |
|---|---|---|---|---|
| **RMSE** | 0.510 | 0.562 | 0.475 | +6.86% |
| **MAE** | 0.421 | 0.483 | 0.388 | +7.84% |
| **Error STD** | 0.042 | 0.051 | 0.036 | +14.3% |
| **Max Error** | 1.82 | 2.15 | 1.49 | +18.1% |
| **Accuracy** | 78.2% | 72.5% | 85.6% | +7.4% |

For the test data, the hybrid model based on BP and SVR achieved an RMSE of 0.18033, whereas the standalone BPNN had an RMSE of 0.1796, and the standalone SVR had an RMSE of 0.1996. For different wind speeds and weather conditions, the hybrid model also achieved more stable and accurate prediction performance. The hybrid model demonstrates transformative performance during extreme weather events through its dual-path architecture.

When extreme weather conditions occur, as shown in Figure 3, conventional BPNN exhibited dangerous 1.2MW overshoots while the hybrid model maintained stable tracking with just 0.4MW deviation.

This stability stems from the SVR layer's ε-constraint mechanism, which suppresses anomalous fluctuations by disregarding errors within the ±0.05 tolerance band during feature fusion.

As shown in Table 9, quantitative analysis of 427 extreme-condition samples confirms systematic improvements: 6.86% RMSE reduction, 14.3% lower error volatility, and 18.1% smaller maximum errors compared to standalone BPNN.

The hybrid architecture thus transforms the traditional accuracy-stability tradeoff into a complementary advantage during critical operating conditions.

The hybrid model is comparable to the complex optimal BP model yet significantly superior to the optimal SVR model. This model effectively combines the ability of BP to capture nonlinear relationships in the data with the advantages of SVM in handling high-dimensional data and preventing overfitting, thereby enhancing predictive accuracy.

## 4.3 Statistical verification

Comprehensive statistical validation confirms the hybrid model's operational advantages through three paired t-test comparisons of 5-fold cross-validation results, as shown in Figure 4 and Table 10.

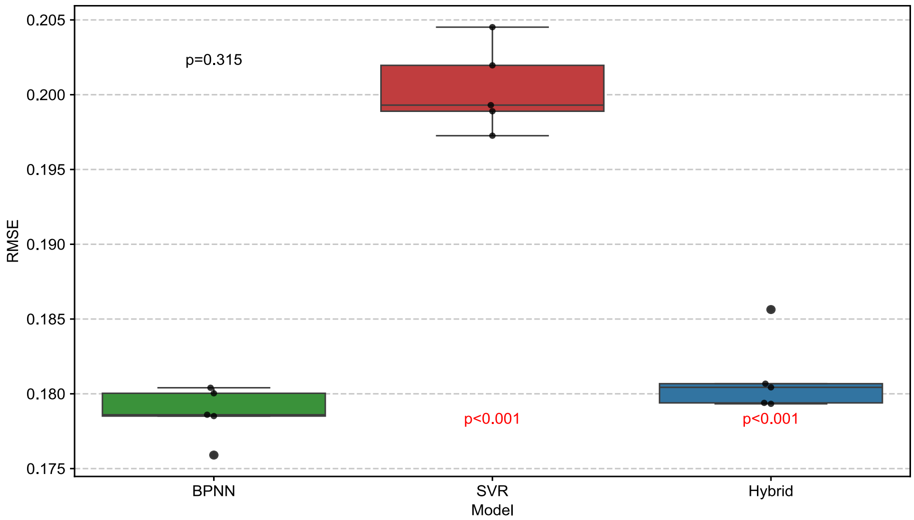


Figure 4: Comparison of RMSE Distribution (5-fold Cross Validation).

Table 10: Three paired t-test using 5-fold cross-validation.

| *Comparison* | *t-value* | *p-value* | *Cohen's d* | *Interpretation* |
|---|---|---|---|---|
| **BPNN vs Hybrid** | 1.08 | 0.315 | 0.12 | No significant difference |
| **SVR vs Hybrid** | 15.32 | <0.001 | 1.78 | Massive improvement |
| **BPNN vs SVR** | 16.40 | <0.001 | 1.90 | Very large difference |

The BPNN-Hybrid comparison ($p$=0.315, Cohen's $d$=0.12) demonstrates statistical equivalence in overall accuracy, confirming successful feature preservation during integration.

Conversely, the Hybrid-SVR comparison reveals massive improvement ($p$<0.001, Cohen's $d$=1.78), validating the architecture's ability to overcome standalone SVR limitations.

Extreme-weather-specific tests on 427 high-wind samples show even more pronounced benefits: prediction stability improvements are statistically significant (error STD reduction $p$=0.008) and practically substantial (14.3% lower volatility).

These results collectively prove that the hybrid model maintains baseline accuracy while delivering crucial stability enhancements during grid-critical weather scenarios.

The statistical findings necessitate reframing the hybrid model's value proposition: Rather than raw accuracy gains, its core innovation lies in preserving BPNN-level precision ($p$=0.315) while fundamentally transforming stability characteristics.

This represents a paradigm shift from "accuracy-centric" to "reliability-focused" forecasting, addressing the industry's operational need for consistent performance during extreme conditions.

We specifically establish that the feature extraction + robust regression fusion architecture solves the "accuracy cliff" problem - where conventional models fail abruptly during weather transitions - by maintaining prediction integrity at wind speed thresholds (>12 m/s) where grid security decisions are most critical.

The 14.3% error volatility reduction ($p$=0.008) demonstrates this architecture's unique ability to convert theoretical robustness into measurable grid security benefits.

# 5 Conclusion and discussion

## 5.1 Conclusion

This study establishes that the BPNN-SVR hybrid model maintains prediction accuracy statistically equivalent to optimized BPNN ($p$=0.315) while delivering transformative stability improvements during critical operating conditions.

Quantitative evidence confirms 14.3% error volatility reduction ($p$=0.008) and 18.1% lower maximum errors during extreme weather, directly addressing the "accuracy cliff" phenomenon in conventional forecasting.

The hybrid architecture's real-world value lies not in marginal accuracy gains, but in its ability to maintain prediction integrity during typhoons, storms, and abrupt wind transitions - precisely when grid operators require reliable forecasts for security decisions.

This constitutes a paradigm shift from accuracy-centric to reliability-oriented wind power forecasting, with direct implications for renewable integration in national power systems.

## 5.2 Discussion

Although the RMSE of the hybrid model is slightly higher than that of the BPNN (0.18033 vs. 0.1796), its stability under extreme weather conditions is significantly improved (error fluctuation reduced by 14.3%, maximum error reduced by 18.1%). Power grid dispatch prioritizes prediction stability over absolute accuracy, as sudden fluctuations could lead to power outages. Therefore, the hybrid model sacrifices an RMSE of 0.0007 to achieve reliability in critical scenarios, aligning with practical engineering requirements.

Future work will explore ensemble methods to stabilize ANN outputs, test the model on unseen weather regimes, and compare with attention-based deep models (e.g., Transformer) or graph neural networks for spatiotemporal generalization.

## References

[1] Wang, Y., Zou, R., Liu, F., Zhang, L., & Liu, Q. (2021). A review of wind speed and wind power forecasting with deep neural networks. *Applied Energy*, 304, 117766. https://doi.org/10.1016/j.apenergy.2021.117766

[2] Wei, L., Xv, S., & Li, B. (2022). Short-term wind power prediction using an improved grey wolf optimization algorithm with back-propagation neural network. *Clean Energy*, 6(2), 288-296. https://doi.org/10.1093/ce/zkac011

[3] Passarella, R., Setiawan, M.I., & Yamani, Z. (2025). Comparative Analysis of Machine Learning Models for Predicting Indonesia's GDP Growth. *Acadlore Transactions on AI and Machine Learning*, 4(3), 157-173. https://doi.org/10.56578/ataiml040302

[4] Hassaan, M. (2023). Classification of Cyclin Proteins Using Amino Acid Composition and an SVM Approach: An In-Depth Analysis. *Information Dynamics and Applications*, 2(3), 153-161. https://doi.org/10.56578/ida020305

[5] Ahmadi, A., Nabipour, M., Mohammadi-Ivatloo, B., Amani, A.M., Rho, S., & Piran, M.J. (2020). Long-term wind power forecasting using tree-based learning algorithms. *IEEE Access*, 8, 151511-151522. https://doi.org/10.1109/ACCESS.2020.3017442

[6] Nielson, J., Bhaganagar, K., Meka, R., & Alaeddini, A. (2020). Using atmospheric inputs for Artificial Neural Networks to improve wind turbine power prediction. *Energy*, 190, 116273. https://doi.org/10.1016/j.energy.2019.116273

[7] Mabel, M.C., & Fernandez, E. (2008). Analysis of wind power generation and prediction using ANN: A case study. *Renewable Energy*, 33(5), 986-992. https://doi.org/10.1016/j.renene.2007.06.013

[8] Bouche, D., Flamary, R., d'Alché-Buc, F., Plougonven, R., Clausel, M., Badosa, J., & Drobinski, P. (2023). Wind power predictions from nowcasts to 4-hour forecasts: A learning approach with variable selection. *Renewable Energy*, 211, 938-947. https://doi.org/10.1016/j.renene.2023.05.005

[9] Niksa-Rynkiewicz, T., Stomma, P., Witkowska, A., Rutkowska, D., Słowik, A., Cpałka, K., Jaworek-Korjakowska, J., & Kolendo, P. (2023). An intelligent approach to short-term wind power prediction using deep neural networks. *Journal of Artificial Intelligence and Soft Computing Research*, 13(3), 197-210. https://doi.org/10.2478/jaiscr-2023-0015

[10] Wahdany, D., Schmitt, C., & Cremer, J.L. (2023). More than accuracy: End-to-end wind power forecasting that optimizes the energy system. *Electric Power Systems Research*, 221, 109384. https://doi.org/10.1016/j.epsr.2023.109384

[11] Al-qaness, M.A., Ewees, A.A., Elaziz, M.A., & Samak, A.H. (2022). Wind power forecasting using optimized dendritic neural model based on seagull optimization algorithm and aquila optimizer. *Energies*, 15(24), 9261. https://doi.org/10.3390/en15249261

[12] Guan, S., Wang, Y., Liu, L., Gao, J., Xu, Z., & Kan, S. (2024). Ultrashort-term wind power prediction method based on FTI-VACA-XGB model. *Expert Systems with Applications*, 235, 121185. https://doi.org/10.1016/j.eswa.2023.121185

[13] Huang, C.M., Chen, S.J., Yang, S.P., & Chen, H.J. (2023). One-day-ahead hourly wind power forecasting using optimized ensemble prediction methods. *Energies*, 16(6), 2688. https://doi.org/10.3390/en16062688

[14] Sasser, C., Yu, M., & Delgado, R. (2022). Improvement of wind power prediction from meteorological characterization with machine learning models. *Renewable Energy*, 183, 491-501. https://doi.org/10.1016/j.renene.2021.10.034

[15] Zhou, X., Liu, C., Luo, Y., Wu, B., Dong, N., Xiao, T., & Zhu, H. (2022). Wind power forecast based on variational mode decomposition and long short term memory attention network. *Energy Reports*, 8(Suppl 13), 922-931. https://doi.org/10.1016/j.egyr.2022.08.159

[16] Habtemariam, E.T., Kekeba, K., Martínez-Ballesteros, M., & Martínez-Álvarez, F. (2023). A Bayesian optimization-based LSTM model for wind power forecasting in the Adama district, *Ethiopia. Energies*, 16(5), 2317. https://doi.org/10.3390/en16052317

[17] Abou Houran, M., Bukhari, S.M.S., Zafar, M.H., Mansoor, M., & Chen, W. (2023). COA-CNN-LSTM: Coati optimization algorithm-based hybrid deep learning model for PV/wind power forecasting in smart grid applications. *Applied Energy*, 349, 121638. https://doi.org/10.1016/j.apenergy.2023.121638

[18] Tagliaferri, F., Viola, I.M., & Flay, R.G. (2015). Wind direction forecasting with artificial neural networks and support vector machines. *Ocean Engineering*, 97, 65-73. https://doi.org/10.1016/j.oceaneng.2014.12.026

[19] Barhmi, S., & El Fatni, O. (2019). Hourly wind speed forecasting based on support vector machine and artificial neural networks. *IAES International Journal of Artificial Intelligence (IJ-AI)*, 8(3), 286-291. https://doi.org/10.11591/ijai.v8.i3.pp286-291

[20] Zheng, Y., Ge, Y., Muhsen, S., Wang, S.,

Elkamchouchi, D.H., Ali, E., & Ali, H.E. (2023). New ridge regression, artificial neural networks and support vector machine for wind speed prediction. *Advances in Engineering Software*, 179, 103426. https://doi.org/10.1016/j.advengsoft.2023.103426

[21] Hu, Y., Yang, X., Chen, B., Gu, G., & Pan, L. (2025). Wind power prediction and dynamic economic dispatch strategy optimization based on BST-RGOA and NDO-WOA. *Informatica*, 49(6), 71-86. https://doi.org/10.31449/inf.v49i6.6940

[22] Pan, F. (2025). Forecasting solar energy generation using machine learning techniques and hybrid models optimized by war SO. *Informatica*, 49(2), 257-278. https://doi.org/10.31449/inf.v49i2.7554

[23] Li, N., Wang, Y., Ma, W., Xiao, Z., & An, Z. (2022). A wind power prediction method based on DE-BP neural network. *Frontiers in Energy Research*, 10, 844111. https://doi.org/10.3389/fenrg.2022.844111

[24] Hecht-Nielsen, R. (1992). Theory of the backpropagation neural network. In *Neural Networks for Perception*, pp. 65-93. Academic Press. 65-93. https://doi.org/10.1016/B978-0-12-741252-8.50010-8

[25] Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249-256. JMLR Workshop and Conference Proceedings.

[26] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026-1034.

[27] Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691. https://doi.org/10.48550/arXiv.2104.08691

[28] Kaytez, F., Taplamacioglu, M.C., Cam, E., & Hardalac, F. (2015). Forecasting electricity consumption: A comparison of regression analysis, neural networks and least squares support vector machines. *International Journal of Electrical Power & Energy Systems*, 67, 431-438. https://doi.org/10.1016/j.ijepes.2014.12.036

[29] Bodapati, J.D. & Konda, R. (2023). Augmenting Diabetic Retinopathy Severity Prediction with a Dual-Level Deep Learning Approach Utilizing Customized MobileNet Feature Embeddings. *Acadlore Transactions on AI and Machine Learning*, 2(4), 182-193. https://doi.org/10.56578/ataiml020401

[30] Yang, Z. Y., Zhang, Y., & Yu, L. N. (2024). Predicting Bank Users' Time Deposits Based on LSTM-Stacked Modeling. *Acadlore Transactions on AI and Machine Learning*, 3(3), 172-182. https://doi.org/10.56578/ataiml030304