energies

MDPI

Article

A Short-Term Wind Power Forecasting Method Based on Multi-Decoder and Multi-Task Learning

Qiang Li [1], Yongzhi Liu [1], Xinyue Yan [2], Haipeng Zhang [3,4], Siyu Wang [1] and Ran Li [2,3,4,*]

[1] Inner Mongolia Power (Group) Co., Ltd., Hohhot 010010, China; mars20200207@163.com (Q.L.); 13314780588@163.com (Y.L.); 13371101000@163.com (S.W.)
[2] College of Smart Energy, Shanghai Jiao Tong University, Shanghai 200240, China; yxy20010513@sjtu.edu.cn
[3] Key Laboratory of Control of Power Transmission and Conversion, Ministry of Education, Shanghai Jiao Tong University, Shanghai 200240, China; sjtu2270150@sjtu.edu.cn
[4] Shanghai Non-Carbon Energy Conversion and Utilization Institute, Shanghai 200240, China
* Correspondence: rl272@sjtu.edu.cn

Abstract

In short-term power forecasting for wind farms, factors such as weather conditions and geographic location lead to certain correlations in the power output of different wind farms, resulting in complex coupling relationships between them. Traditional wind power forecasting methods often predict each wind farm independently, without considering these coupling relationships. To address this issue, this paper proposes a multi-task Transformer model based on multiple decoders, which accounts for the intrinsic connections between different wind farms, enabling joint power forecasting across multiple sites. The proposed model adopts a single encoder-multiple decoder structure, where a unified encoder processes all input data, and multiple decoders perform prediction tasks for each wind farm separately. Testing on actual wind farm data from the Inner Mongolia region of China shows that, compared to other forecasting models, the proposed model significantly improves the accuracy of power predictions for different wind farms.

Keywords: renewable energy forecasting; multi-decoder; Transformer model; attention mechanism; deep learning

1. Introduction

Wind energy, as a clean, renewable, and abundant energy source [1], plays an important role in new energy power generation. However, owing to the pronounced randomness and volatility of wind [2], the power output of wind farms exhibits considerable uncertainty [3]. This uncertainty not only poses challenges to the stable operation of power systems but also affects grid dispatching and electricity market transactions. Therefore, accurate wind power forecasting is of great significance for improving the quality of wind power integration [4], optimizing power system operation [5], and enhancing the economic efficiency of electricity markets [6].

Conventional wind power forecasting methods typically perform independent forecasting for different wind farms. Typical methods for short-term wind power forecasting include time-series extrapolation [7], long short-term memory (LSTM) networks [8], extreme learning machines (ELM) [9], and others. In [10], a wind power forecasting model based on a convolutional neural network (CNN)–LSTM architecture is proposed, which improves the forecasting accuracy of wind farm power output in the planning stage when historical operational data are scarce. In [11], the forecasting results of multiple models,

Check for updates

Academic Editor: Francesco Castellani

Received: 1 December 2025
Revised: 27 December 2025
Accepted: 29 December 2025
Published: 10 January 2026

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

including support vector machine (SVM), multilayer perceptron (MLP), and recurrent neural network (RNN), are integrated to further enhance forecasting accuracy. However, due to factors such as geographical correlations, the power outputs of different wind farms exhibit complex coupling relationships. If such coupling can be further taken into account, the accuracy of wind power forecasting at the farm level can be improved to some extent.

The multi-output structure of artificial neural networks can meet the needs of multiple forecasting tasks. In [12], a multitask learning-based load forecasting method is proposed, in which CNN and gated recurrent unit (GRU) networks serve as shared layers to capture the coupling among multiple loads, while a gradient boosting regression tree (GBRT) model is employed to perform joint multivariate load forecasting. In [13], a short-term multivariate load forecasting model based on GRU networks is proposed. By modifying the gating structure of GRU, a multi-level gated recurrent architecture is constructed to achieve multivariate load forecasting. However, as recurrent neural networks, CNN and GRU have inherent limitations: they are prone to gradient vanishing when dealing with time-series data and have difficulty capturing very long-term dependencies.

The Transformer model, first proposed by Vaswani et al. in 2017 with attention mechanisms at its core [14], has achieved great success in fields such as natural language processing, image analysis, and video understanding. In the field of wind power forecasting, when dealing with highly complex and stochastic input sequences of renewable power data, the Transformer can dynamically adjust its focus on meteorological factors such as wind speed and wind direction through the self-attention mechanism. It can capture the correlations among key variables at any temporal and spatial positions and assign higher weights to more informative signals [15], thereby improving the accuracy of wind power forecasting. In recent years, Transformer models based on attention mechanisms have been shown to outperform recurrent neural network models on many tasks. Thanks to their parallel computation capability, Transformers exhibit better gradient flow during training and avoid the gradient vanishing problem commonly encountered in recurrent neural networks. Moreover, the self-attention mechanism can directly attend to the relationships between any two positions in a sequence, making Transformers more efficient in capturing long-range dependencies. Therefore, compared with LSTM and other recurrent neural networks, Transformer models offer a broader range of dependencies and stronger generalization capability.

Existing Transformer-based wind power forecasting approaches rarely adopt a multi-decoder/multi-task design to jointly model multiple spatially correlated wind farms. Instead, these models are typically built independently for each wind farm, with each Transformer model containing only a single decoder. As a result, they fail to explicitly account for the wake effects and spatial dependencies among different wind farms, which limits their forecasting accuracy. This paper addresses this gap by proposing a novel approach with the following key contributions.

(1) We propose a multi-decoder, multitask learning model for short-term wind power forecasting, based on the Transformer architecture. The proposed model adopts a single-encoder–parallel-decoder structure, enabling joint forecasting of power outputs for multiple wind farms.
(2) We design a unified encoder to map the input data into a latent representation matrix. Attention mechanisms are employed to extract high-dimensional interaction features from the correlated information across multiple farms. Each farm-specific decoder then combines the latent representation matrix with the input features specific to its forecasting task, enabling accurate predictions for each farm.

(3) We conduct extensive experiments, including case studies using real wind farm data from Inner Mongolia, China. These experiments demonstrate that the proposed model significantly improves the accuracy of wind power forecasting.

2. Multi-Decoder and Multi-Task Learning Architecture

The overall architecture of the multi-decoder, multitask learning Transformer is shown in Figure 1. It is characterized by a single encoder and multiple task-specific decoders.

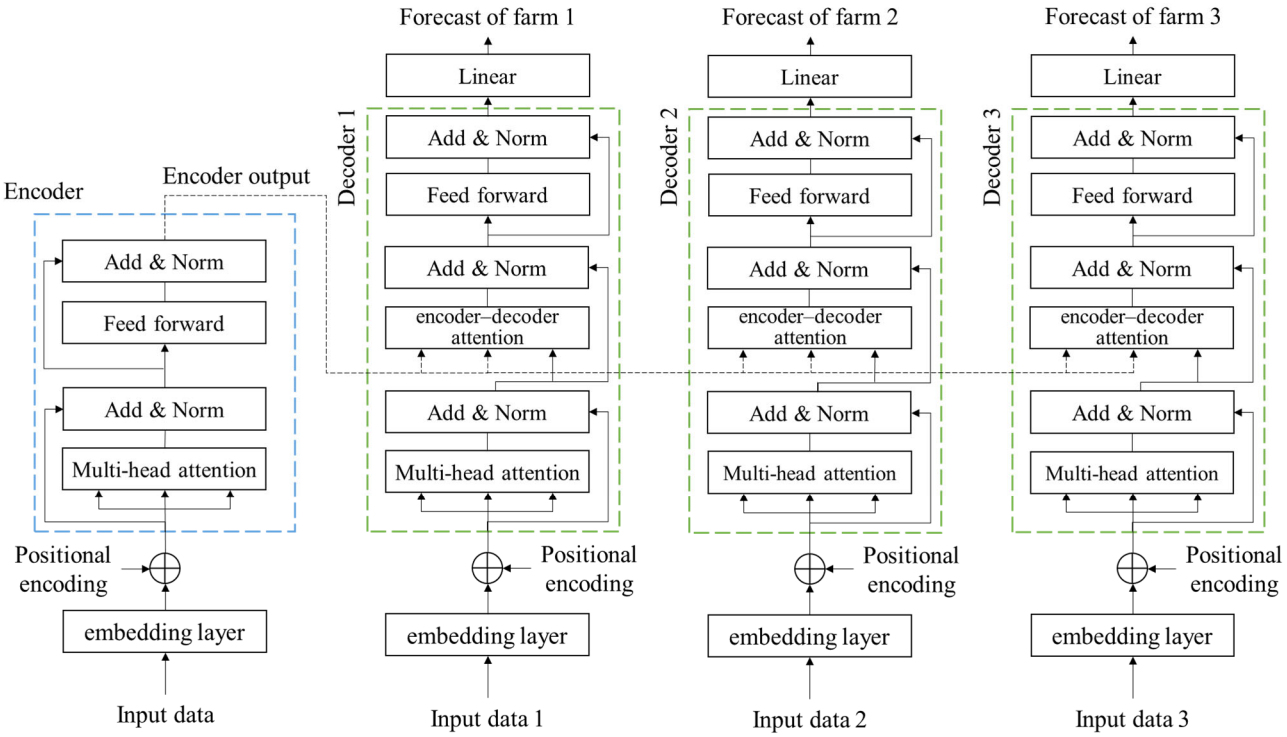


Figure 1. Architecture of the multi-decoder and multi-task learning model.

In the Transformer model, meteorological data for all wind farms—including the wind speed components in the latitude and longitude directions at 100 m and 10 m, surface pressure, and 2 m temperature—are first passed through an embedding layer and then fed into the encoder. The embedding layer maps the input data into a new feature space via linear transformation or feature extraction, while positional encoding is used to add temporal information to the time-series data. The encoder mainly consists of a multi-head self-attention mechanism, a feedforward neural network, and residual connections with layer normalization. The multi-head self-attention mechanism mimics the resource allocation mechanism of human attention by assigning higher probabilities to important information in the sequence, thereby highlighting key information and reducing or even completely ignoring unimportant information. The feedforward neural network enhances the nonlinear fitting capability of the model, whereas normalization accelerates the convergence of model training.

The structure of each decoder is similar to that of the encoder and mainly includes an embedding layer, a multi-head self-attention mechanism, an encoder–decoder attention mechanism, a feedforward neural network, residual connections with layer normalization, and a final fully connected layer. The input data of each decoder consists of the meteorological variables corresponding to its associated wind farm, such as the wind speed components in the latitude and longitude directions at 100 m and 10 m, surface pressure, and 2 m temperature. The embedding layer maps the input data into a new feature space, while positional encoding adds temporal information to the time-series data. It is worth

noting that the second multi-head attention module in the decoder is not a self-attention mechanism. In this module, the keys and values come from the encoder output, whereas the queries come from the output of the first multi-head attention module in the decoder. In this way, each decoder can fully exploit the encoded information during the decoding process. Finally, the fully connected layer transforms the decoder outputs into the final forecasting results.

The model performs a day-ahead short-term wind power forecasting task. At 08:00 on day D, the model takes the most recent available historical observations as inputs and generates the forecast profile for the entire next day, covering 00:00 to 24:00 on day $D + 1$ at a 15 min resolution, resulting in 96 forecasting points per day. The test period is evaluated under a static-origin day-ahead forecasting protocol, and direct multi-step forecasting is adopted.

3. Transformer-Based Wind Power Forecasting Model

The wind power forecasting model proposed in this paper is built on the Transformer architecture. This section presents in detail the basic principles and mathematical formulations of positional encoding, the multi-head attention mechanism, layer normalization, and the feedforward neural network used in the model.

3.1. Positional Encoding

In the multi-head self-attention mechanism, the attention scores are computed only based on the overall information of the input data, while the positional information of the input sequence is ignored. To address this issue, positional encoding is introduced to provide the Transformer with the position of each input element [16], thereby improving the prediction performance of the model when processing time-series data.

For the element at the k-th dimension and the p-th position in the input sequence, the positional encoding $PE(p, k)$ is defined as follows:

$$PE(p, 2k) = \sin\left(\frac{p}{10000^{2k/d}}\right) \tag{1}$$

$$PE(p, 2k+1) = \cos\left(\frac{p}{10000^{2k/d}}\right) \tag{2}$$

The positional encoding not only provides the absolute positional information of different elements, but also exploits the periodicity of sine and cosine functions (i.e., $PE(p + p', k)$ can be expressed as a linear transformation of $PE(p, k)$) to encode the relative positional relationships of the input data.

3.2. Attention Mechanism

In renewable energy power forecasting, it is necessary to extract both temporal and feature information of uncertain quantities such as wind speed, temperature and humidity, which makes the input information highly complex. The multi-head self-attention mechanism can adaptively focus on different parts of the input, capture rich feature information, and assign higher weights to key information. Therefore, in this paper, the encoder is built around the multi-head self-attention mechanism to extract the feature information of each uncertain variable and its related variables, ensuring that the model can focus on the most important parts of these uncertain features [17].

The multi-head self-attention mechanism characterizes the correlations between each data point and all other data points in the sequence by computing attention scores. In this paper, dot-product attention is adopted. Let the input data be denoted by $\boldsymbol{x} \in \mathbb{R}^{n \times d}$, where n is the number of samples and d is the feature dimension. First, $\boldsymbol{x}$ is linearly transformed

to generate the query, key and value matrices. Using the corresponding weight matrices, the linear transformations of $\boldsymbol{x}$ are given by (3)–(5):

$$\boldsymbol{Q}_i = \boldsymbol{x}\boldsymbol{W}_i^Q \tag{3}$$

$$\boldsymbol{K}_i = \boldsymbol{x}\boldsymbol{W}_i^K \tag{4}$$

$$\boldsymbol{V}_i = \boldsymbol{x}\boldsymbol{W}_i^V \tag{5}$$

where for the i-th attention head, the query matrix, key matrix and value matrix are denoted by $\boldsymbol{Q}_i$, $\boldsymbol{K}_i$ and $\boldsymbol{V}_i$, respectively, each of dimension $n \times d_k$. The matrices $\boldsymbol{W}^Q$, $\boldsymbol{W}^K$ and $\boldsymbol{W}^V$ are the corresponding weight matrices of dimension $d \times d_k$, which map the input features from a $n \times d$-dimensional space to a $n \times d_k$-dimensional space.

The correlation between each query in the query matrix and all keys in the key matrix is computed via the dot product, and the dot product results are scaled to avoid excessively large values caused by high dimensionality. A softmax function is then applied to the scaled dot products to obtain the attention weight matrix:

$$\boldsymbol{\alpha}_i = \text{softmax}\left(\frac{\boldsymbol{Q}_i \boldsymbol{K}_i^{\mathrm{T}}}{\sqrt{d_k}}\right) \tag{6}$$

where $\boldsymbol{\alpha}_i$ denotes the attention weight matrix of the i-th head.

By multiplying the attention weight matrix with the value matrix, the attention output matrix is obtained:

$$\boldsymbol{A}_i = \boldsymbol{\alpha}_i \boldsymbol{V}_i \tag{7}$$

where $\boldsymbol{A}_i$ denotes the attention output matrix of the i-th head.

By aggregating the outputs of multiple attention heads, the final output of the multi-head attention mechanism is obtained:

$$\text{Attention}(\boldsymbol{x}) = [\boldsymbol{A}_1\, \boldsymbol{A}_2\, \cdots\, \boldsymbol{A}_h]\boldsymbol{W}^O \tag{8}$$

where Attention($\boldsymbol{x}$) is the final output of the multi-head attention module; h is the number of attention heads; and $\boldsymbol{W}^O$ is a linear projection matrix that maps the concatenated vector back to the $n \times d$-dimensional space.

In the decoder, in addition to the multi-head self-attention mechanism, an encoder–decoder attention mechanism is used. The encoder–decoder attention mechanism utilizes the contextual information generated by the encoder to enable more accurate decoding. Compared with the multi-head self-attention mechanism, the query matrix of the encoder–decoder attention comes from the output of the self-attention layer in the decoder and represents the target sequence information, denoted by $\boldsymbol{Q}^D$; the key and value matrices come from the intermediate representations output by the encoder and represent the feature sequence information, denoted by $\boldsymbol{K}^E$ and $\boldsymbol{V}^E$, respectively, as shown in (9):

$$\boldsymbol{\alpha}_i^{\text{E-D}} = \text{softmax}\left(\frac{\boldsymbol{Q}_i^D (\boldsymbol{K}_i^E)^{\mathrm{T}}}{\sqrt{d_k}}\right) \tag{9}$$

3.3. Layer Normalization and Feedforward Neural Network

In the Transformer model, residual connections and layer normalization are typically used together. Residual connections directly add the input of a layer to its output so that the layer learns the residual with respect to the input, which helps mitigate the vanishing and exploding gradient problems in deep neural networks. Let $\boldsymbol{x}^{(0)}$ and output($\boldsymbol{x}^{(0)}$) denote

the input and output of a layer before the residual connection, respectively. The output of the residual connection, $\boldsymbol{x}^{(1)}$, is given by:

$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} + \text{output}(\boldsymbol{x}^{(0)}) \tag{10}$$

where + denotes element-wise addition.

The output of the residual connection then serves as the input to the layer normalization. Layer normalization normalizes all features of each sample so that the output of each layer has zero mean and unit variance, thereby accelerating model convergence. The computation of layer normalization is given by (11)–(13):

$$\mu_i = \frac{1}{d}\sum_{j=1}^{d} x_{ij}^{(1)} \tag{11}$$

$$\sigma_i^2 = \frac{1}{d}\sum_{j=1}^{d} (x_{ij}^{(1)} - \mu_i)^2 \tag{12}$$

$$x_{ij}^{(2)} = \gamma_j \frac{x_{ij}^{(1)} - \mu_i}{\sqrt{\sigma_i^2 + \varepsilon}} + \beta_j \tag{13}$$

where μ_i and σ_i^2 denote the mean and variance of the i-th sample, respectively; ε is a constant introduced to prevent division by zero; and γ_j and β_j are learnable parameters for each feature dimension. The output of layer normalization is denoted by $\boldsymbol{x}^{(2)} = \{x_{ij}^{(2)} \mid i = 1, 2, \ldots, n; j = 1, 2, \ldots, d\}$.

The position-wise feedforward neural network is an important component of both the encoder and each decoder in the Transformer model. Its purpose is to process the input data through two linear transformations with a nonlinear activation function in between, thereby extracting more complex feature information. The input to the feedforward neural network is typically the output of layer normalization, denoted by $\boldsymbol{x}^{(2)}$, and its output can be expressed as:

$$\text{FFN}(\boldsymbol{x}^{(2)}) = \text{ReLU}(\boldsymbol{x}^{(2)}\boldsymbol{W}_1^L + \boldsymbol{b}_1)\boldsymbol{W}_2^L + \boldsymbol{b}_2 \tag{14}$$

where $\text{FFN}(\boldsymbol{x}^{(2)})$ denotes the output of the feedforward neural network; ReLU(.) = max(0, .); $\boldsymbol{W}_1^L$ and $\boldsymbol{W}_2^L$ are the weight matrices of the linear transformations; $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are the corresponding bias vectors.

4. Case Study Analysis

4.1. Dataset Description

In this case study, the power outputs of three wind farms located in the Inner Mongolia region are forecasted. The historical operational data of the wind farms and the numerical weather prediction data used in this paper are all collected from wind farms in the Inner Mongolia Autonomous Region.

The meteorological input features of the forecasting model include wind speed, wind direction, temperature, humidity, and air pressure. The training dataset covers the period from May 2023 to November 2023, and the testing dataset spans December 2023, resulting in an 87%/13% split between training and testing. The temporal resolution of the data is 15 min. Prior to model training, abnormal samples are removed and missing values are imputed. Min–Max normalization is then applied during both training and inference.

The model is trained using the Adam optimizer with mean squared error (MSE) as the training objective. For multi-task learning across multiple wind farms, the overall loss is formulated as a weighted sum of per-farm MSE losses; equal task weights are

adopted in this work. The initial learning rate is 0.001 and is decayed using a step-decay schedule during training. The model is trained for 200 epochs with a batch size of 20, and early stopping is applied when the MSE does not improve for 10 consecutive epochs. The proposed model has 3.0 M trainable parameters, with an average training time of approximately 50 s per epoch on a single NVIDIA RTX 4060 GPU (NVIDIA Corporation, Santa Clara, CA, USA).

4.2. Evaluation Metrics

The evaluation metrics adopted in this case study are R^2 (R-squared), root mean square error (RMSE), mean absolute error (MAE), and normalized root mean square error (nRMSE). Their corresponding formulations are given in (15)–(18).

$$R^2 = 1 - \frac{\sum_{t=1}^{T} (\hat{y}_t - y_t)^2}{\sum_{t=1}^{T} \left(\sum_{t=1}^{T} \frac{y_t}{T} - y_t \right)^2} \tag{15}$$

$$\text{RMSE} = \sqrt{\frac{1}{T} \sum_{t=1}^{T} (\hat{y}_t - y_t)^2} \tag{16}$$

$$\text{MAE} = \frac{1}{T} \sum_{t=1}^{T} |\hat{y}_t - y_t| \tag{17}$$

$$\text{nRMSE} = \sqrt{\frac{1}{T} \sum_{t=1}^{T} \left(\frac{\hat{y}_t - y_t}{C_t} \right)^2} \tag{18}$$

where T is the total number of test samples; t is the sample index; $\hat{y}_t$ is the predicted value of the t-th test sample; y_t is the true value of the t-th test sample; and C_t denotes the installed capacity of the wind farm for the t-th test sample.

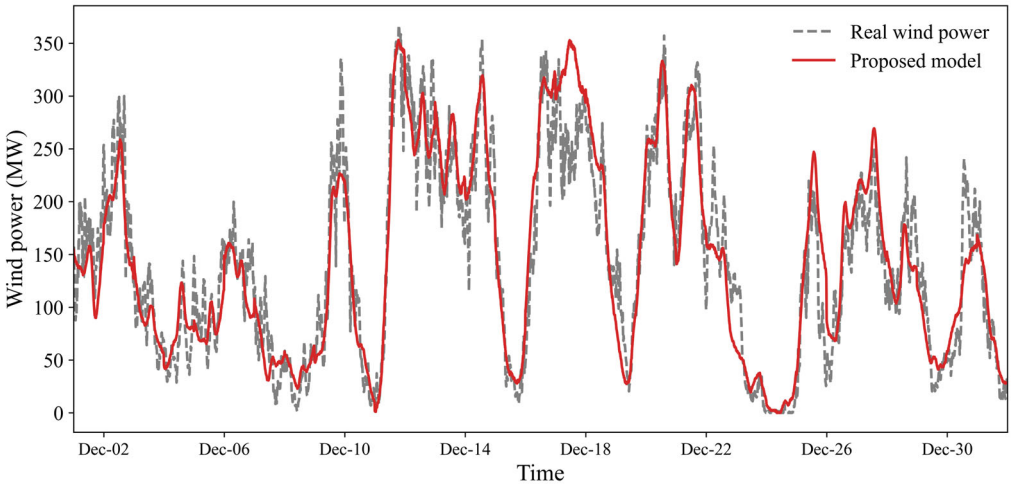
4.3. Comparison with the Independently Trained Transformer Model

In the proposed multi-decoder multi-task learning Transformer model, the number of decoders is fixed to match the number of wind farms (one decoder per wind farm). The number of attention heads is set to 4 and adopted as the default setting in this work, as it provides a good accuracy–cost trade-off validated experimentally. The number of encoder layers and decoder layers is set to 1, and their hidden dimension is 256 for both. To prevent overfitting, the dropout rate is set to 0.1 throughout all layers.

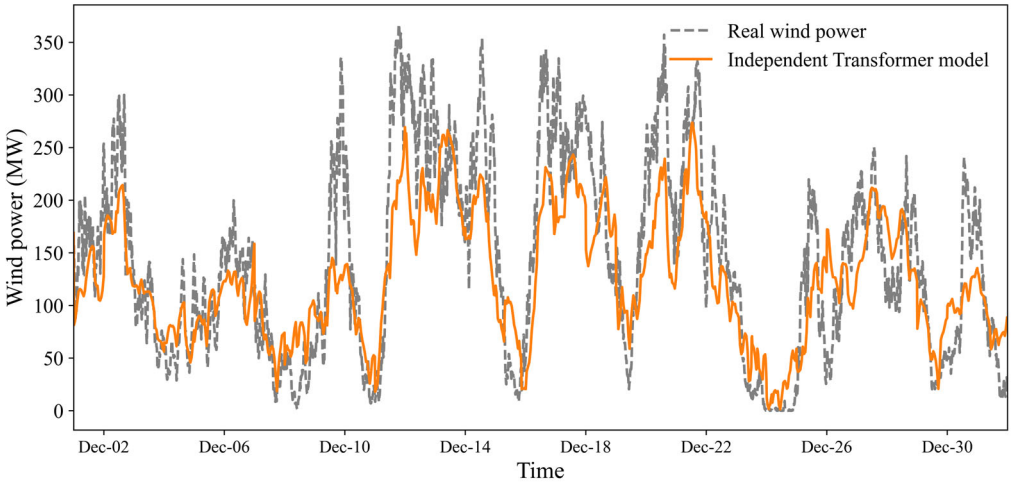
The proposed model is compared with a Transformer model that independently forecasts each wind farm. The indicator analysis is conducted under four scenarios:

- Scenario 1 evaluates the forecasting performance for wind farm 1,
- Scenario 2 evaluates the forecasting performance for wind farm 2,
- Scenario 3 evaluates the forecasting performance for wind farm 3, and
- Scenario 4 evaluates the metrics based on the total power output of wind farms 1, 2, and 3.

Taking Scenario 4 as an example, the forecasting results of the proposed model and the independent Transformer model are shown in Figure 2.



(**a**)



(**b**)

Figure 2. Forecasting results of the proposed model and the independent Transformer model. (**a**) Forecasting results of the proposed model, which performs joint forecasting for multiple wind farms using a multi-decoder, multitask learning architecture. (**b**) Forecasting results of the independent Transformer model, which builds separate Transformer models to forecast each wind farm individually.

The evaluation metric results, such as R^2, RMSE, MAE, and nRMSE for the proposed model and the independent Transformer model, are shown in Table 1.

Table 1. Comparison with an independently predicted transformer model.

Scenario	Model	R^2	RMSE	MAE	nRMSE
1	Proposed model	0.60	23.76	17.59	0.14
	Independent Transformer model	0.35	30.51	25.49	0.19
2	Proposed model	0.63	7.43	5.70	0.15
	Independent Transformer model	0.35	9.74	7.40	0.20
3	Proposed model	0.86	23.26	16.33	0.12
	Independent Transformer model	0.55	41.14	30.55	0.20
4	Proposed model	0.87	33.52	25.25	0.09
	Independent Transformer model	0.63	55.90	44.00	0.17

According to the data in Table 1, the proposed model achieves significant improvements over the independent Transformer model across all evaluation metrics. In terms of R^2, in Scenario 1, the proposed model attains an R^2 of 0.60, whereas the independent Transformer model achieves only 0.35, corresponding to an improvement of 71.4%. In Scenario 2, the R^2 of the proposed model is 0.63, which is 80.0% higher than that of the independent Transformer model (0.35). In Scenario 3, the proposed model reaches an R^2 of 0.86, compared with 0.55 for the independent Transformer model, representing an improvement of 56.4%. In Scenario 4, the proposed model obtains an R^2 of 0.87, while the independent Transformer model achieves 0.63, yielding an improvement of 38.1%. These results indicate that the proposed model can better fit the data and enhance the reliability of the forecasts.

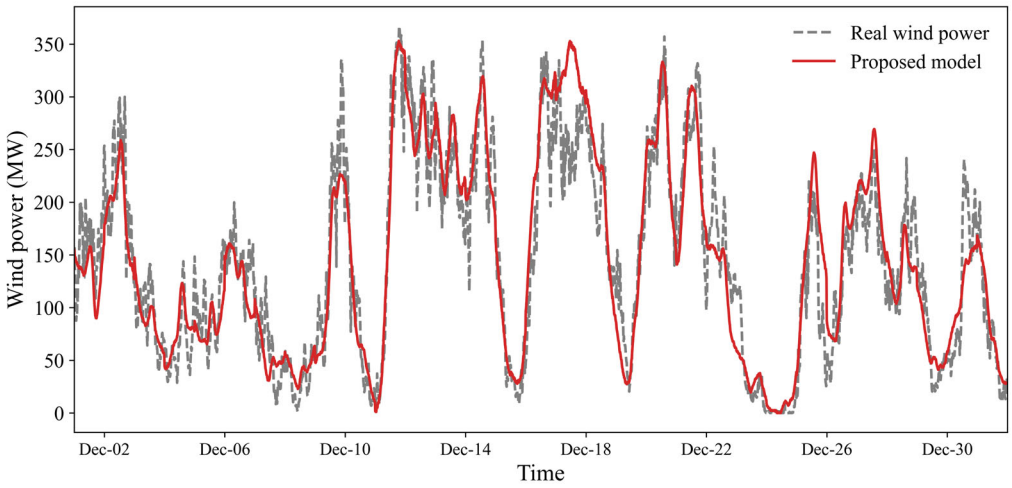
Regarding RMSE and MAE, the proposed model outperforms the independent Transformer model in all scenarios, indicating a substantial reduction in prediction errors. In Scenario 1, the RMSE of the proposed model is 23.76, compared with 30.51 for the independent Transformer model, corresponding to a 22.1% reduction in error; meanwhile, MAE decreases from 25.49 to 17.59, a reduction of 31.0%. In Scenario 2, the RMSE of the proposed model is 7.43, compared with 9.74 for the independent Transformer model, representing a 23.7% decrease; MAE decreases from 7.40 to 5.70, a reduction of 23.0%. In Scenario 3, the RMSE of the proposed model is 23.26, while that of the independent Transformer model is 41.14, corresponding to a 43.5% reduction; MAE decreases from 30.55 to 16.33, a reduction of 46.5%. In Scenario 4, the RMSE of the proposed model is 33.52, compared with 55.90 for the independent Transformer model, yielding a 40.0% reduction; MAE decreases from 44.00 to 25.25, a reduction of 42.6%.

In terms of nRMSE, which is a normalized metric, in Scenario 1, the nRMSE of the proposed model is 0.14, whereas the independent Transformer model achieves 0.19, corresponding to a 26.3% improvement. In Scenario 2, the nRMSE of the proposed model is 0.15, compared to 0.20 for the independent Transformer model, resulting in a 25.0% improvement. In Scenario 3, the proposed model achieves an nRMSE of 0.12, while the independent Transformer model reaches 0.20, representing a 40.0% improvement. In Scenario 4, the nRMSE of the proposed model is 0.09, whereas that of the independent Transformer model is 0.13, still yielding a 30.8% improvement.

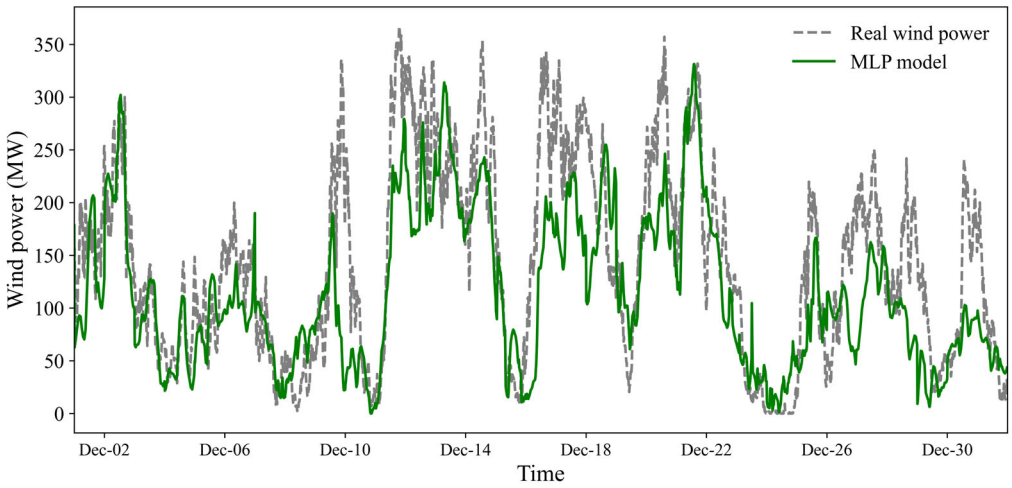
Overall, the proposed model performs better than the independent Transformer model across all evaluation metrics: it improves R^2 by 38.1–80.0%, reduces RMSE by 22.1–43.5%, reduces MAE by 23.0–46.5%, and reduces nRMSE by 25.0–40.0%. These results demonstrate that the proposed model surpasses the independent Transformer model in terms of fitting capability, prediction accuracy, and stability, and thus exhibits superior forecasting performance.

4.4. Comparison with Joint Forecasting Deep Learning Models

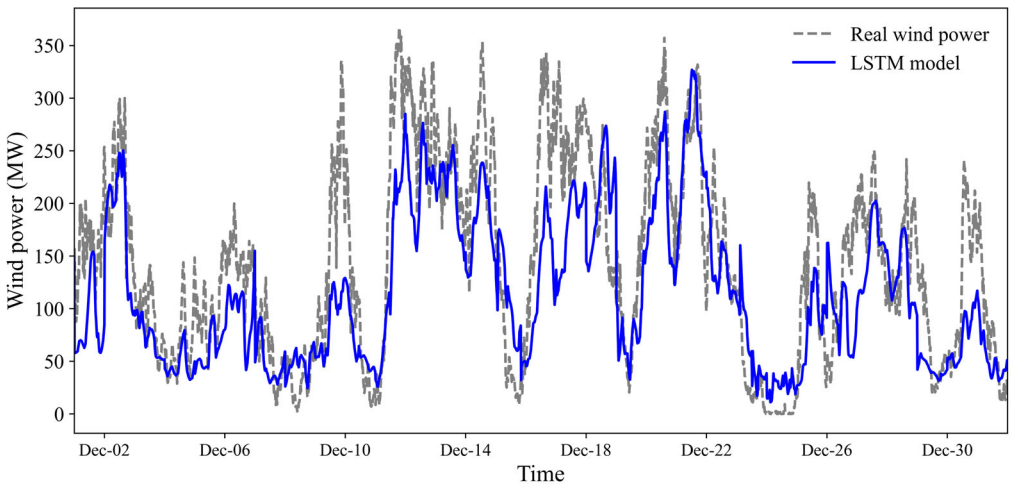
The proposed multi-decoder, multitask learning Transformer model is further compared with machine learning models that perform joint forecasting for all wind farms. The baseline machine learning models considered in this paper include an MLP model, an LSTM model, and a CNN model. All three baseline models are implemented using standard Python libraries (Python 3.10.1) and trained using the Adam optimizer with MSE loss. Their architectures and hyperparameters are selected via grid search. The forecasting result curves of each model are shown in Figure 3.



(**a**)

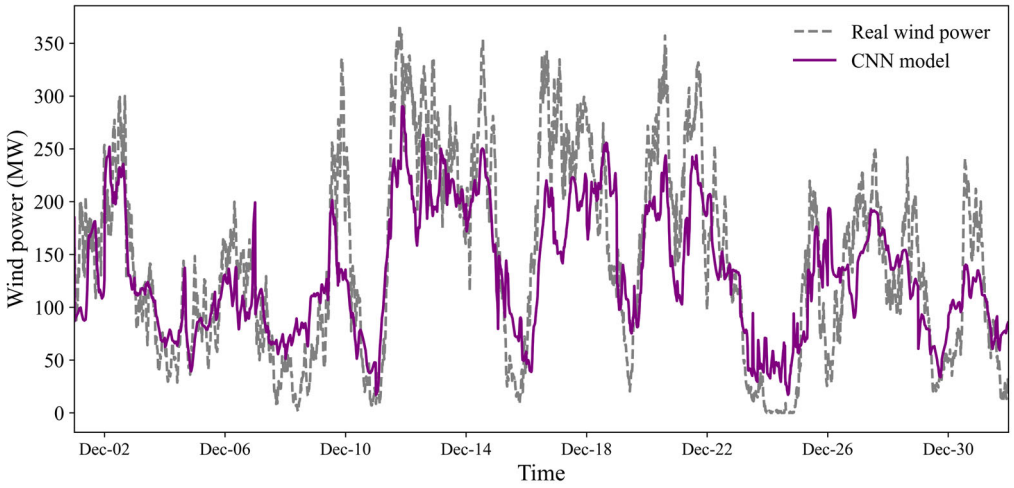


(**b**)



(**c**)

Figure 3. *Cont.*



(**d**)

Figure 3. Forecasting results of the proposed model, MLP model, LSTM model, and CNN model. (**a**) Forecasting results of the proposed model. (**b**) Forecasting results of the MLP model. (**c**) Forecasting results of the LSTM model. (**d**) Forecasting results of the CNN model.

The evaluation metric results, such as R^2, RMSE, MAE, and nRMSE for the models are shown in Table 2. These results indicate that the proposed model demonstrates clear advantages over the other joint forecasting machine learning models. In terms of key performance metrics such as R^2, RMSE, MAE, and nRMSE, the proposed model consistently outperforms the MLP, LSTM, and CNN models, highlighting its superiority in the forecasting task.

Table 2. Comparison with joint prediction machine learning models.

Scenario	Model	R^2	RMSE	MAE	nRMSE
1	Proposed model	0.60	23.76	17.59	0.14
	MLP model	0.23	33.17	26.48	0.20
	LSTM model	0.25	32.82	26.31	0.20
	CNN model	0.30	31.50	25.21	0.19
2	Proposed model	0.63	7.43	5.70	0.15
	MLP model	0.23	12.35	9.29	0.25
	LSTM model	0.28	10.26	8.42	0.21
	CNN model	0.14	15.54	11.67	0.31
3	Proposed model	0.86	23.26	16.33	0.12
	MLP model	0.40	47.38	34.52	0.24
	LSTM model	0.48	44.06	31.75	0.22
	CNN model	0.51	42.96	34.08	0.21
4	Proposed model	0.87	33.52	25.25	0.09
	MLP model	0.51	64.07	48.53	0.15
	LSTM model	0.55	61.13	47.07	0.15
	CNN model	0.63	55.90	45.11	0.13

In terms of R^2, the proposed model outperforms all other models in every scenario. In Scenario 1, the proposed model achieves an R^2 of 0.60, which is 100% higher than that of the best-performing CNN model and 160.9% higher than that of the worst-performing MLP model. In Scenario 2, the R^2 of the proposed model is 0.63, representing an improvement of 125.0% over the best-performing LSTM model and 350.0% over the worst-performing CNN model. In Scenario 3, the proposed model attains an R^2 of 0.86, which is 68.6% higher than that of the best-performing CNN model and 115.0% higher than that of the worst-performing MLP model. In Scenario 4, the proposed model reaches an R^2 of 0.87,

exceeding the best-performing CNN model by 38.1% and the worst-performing MLP model by 70.6%.

With respect to RMSE and MAE, the proposed model also shows a clear advantage, effectively reducing prediction errors and improving forecasting accuracy. In Scenario 1, the RMSE of the proposed model is 23.76, which is 24.6% lower than that of the best-performing CNN model and 28.4% lower than that of the worst-performing MLP model. Meanwhile, the MAE is 17.59, representing reductions of 30.2% and 33.6% compared with the best-performing CNN model and the worst-performing MLP model, respectively. In Scenario 2, the proposed model achieves an RMSE of 7.43, which is 27.6% lower than that of the best-performing LSTM model and 52.2% lower than that of the worst-performing CNN model; the MAE is 5.70, 32.3% lower than that of the best-performing LSTM model and 51.2% lower than that of the worst-performing CNN model. In Scenario 3, the RMSE of the proposed model is 23.26, corresponding to reductions of 45.9% and 50.9% relative to the best-performing CNN model and the worst-performing MLP model, respectively; the MAE is 16.33, 48.6% lower than that of the best-performing LSTM model and 52.7% lower than that of the worst-performing MLP model. In Scenario 4, the proposed model attains an RMSE of 33.52, which is 40.0% lower than that of the best-performing CNN model and 47.7% lower than that of the worst-performing MLP model; the MAE is 25.25, representing reductions of 44.0% and 47.9% compared with the best-performing and worst-performing CNN and MLP models, respectively.

In terms of nRMSE, the proposed model consistently outperforms all other models in every scenario. In Scenario 1, the nRMSE of the proposed model is 0.14, which is 26.3% lower than that of the best-performing CNN model and 30.0% lower than that of the worst-performing MLP and LSTM models. In Scenario 2, the proposed model achieves an nRMSE of 0.15, improving upon the best-performing LSTM model by 28.6% and the worst-performing CNN model by 51.6%. In Scenario 3, the nRMSE of the proposed model is 0.12, which is 42.9% lower than that of the best-performing CNN model and 50.0% lower than that of the worst-performing MLP model. In Scenario 4, the proposed model reaches an nRMSE of 0.09, exceeding the best-performing CNN model and the worst-performing MLP and LSTM models by 30.8% and 40.0%, respectively.

Overall, the proposed model consistently outperforms other common models such as MLP, LSTM, and CNN in all scenarios, with significant improvements in the key metrics R^2, RMSE, MAE, and nRMSE. The improvement in R^2 ranges from 70.6% to 173.9%, with Scenario 2 showing the most pronounced gain and Scenario 4 exhibiting a relatively smaller increase. The reduction in RMSE lies between 28.4% and 49.0%, with the largest error reduction observed in Scenario 3 and the smallest in Scenario 1. The decrease in MAE ranges from 32.3% to 53.7%, where Scenario 3 achieves the greatest reduction and Scenario 2 the smallest. The reduction in nRMSE ranges from 26.3% to 51.6%, with the highest gain in Scenario 2 and a relatively smaller improvement in Scenario 1. These results demonstrate that the proposed model exhibits significant performance advantages across different scenarios.

4.5. Overall Comparison of All Models

To enable a horizontal comparison of all forecasting models, this section takes Scenario 4 as an example and compares the statistical forecasting metrics of different models, including R^2, RMSE, MAE, and nRMSE, to provide a more intuitive view of their performance. In addition, to reflect the improvement brought by the proposed forecasting model in terms of power-system economic benefits, the daily operating cost is introduced as a cost metric. Based on the IEEE 30-bus test system, a power-system operation workflow is simulated [18], including day-ahead unit commitment and intra-day economic dispatch. The cost metric is

defined as the daily average total operating cost of the system, i.e., the sum of the day-ahead cost and the intra-day cost. The thermal unit parameters are listed in Table 3.

Table 3. Parameters of the thermal generating units.

Unit ID	Rated Capacity (MW)	Energy Price (CNY/MW)	Upward Reserve Capacity (MW)	Downward Reserve Capacity (MW)	Upward Reserve Price (CNY/MW)	Downward Reserve Price (CNY/MW)
1	10,205	500	5000	5000	1200	1200
2	6480	480	2800	2800	1000	1000
3	3888	550	1600	1600	1250	1250
4	5184	460	2400	2400	1000	1000
5	2592	510	1200	1200	1200	1200
6	2592	500	1120	1120	1150	1150

As shown in Table 4, the overall comparison across all models indicates that the forecasting performance of the Independent Transformer model and other joint-forecasting models (MLP model, LSTM model, and CNN model) is broadly comparable, whereas the proposed model achieves a clear improvement in forecasting accuracy and yields enhanced economic benefits for power system operation. In terms of accuracy metrics, the proposed model improves R^2 by 38.10% compared with the Independent Transformer model and by 70.59%, 58.18%, and 38.10% compared with the MLP, LSTM, and CNN models, respectively. For RMSE, the proposed model improves performance by 40.04% relative to the Independent Transformer model and by 47.68%, 45.17%, and 40.04% relative to the MLP, LSTM, and CNN models, respectively. For MAE, the corresponding improvements are 42.61% over the Independent Transformer model and 47.97%, 46.36%, and 44.03% over the MLP, LSTM, and CNN models, respectively. For the normalized metric nRMSE, the proposed model achieves improvements of 30.77% over the Independent Transformer model and 40.00%, 40.00%, and 30.77% over the MLP, LSTM, and CNN models, respectively. In terms of the cost metric, the proposed model reduces the daily average total operating cost by 5.54% compared with the Independent Transformer model and by 11.37%, 9.50%, and 7.29% compared with the MLP, LSTM, and CNN models, respectively, thereby effectively lowering system operating costs and improving economic efficiency.

Table 4. Comparison of forecasting performance across all models under Scenario 4 in terms of R^2, RMSE, MAE, nRMSE, and cost.

Scenario	Model	R^2	RMSE	MAE	nRMSE	Cost/(CNY/Day)
4	Proposed model	0.87	33.52	25.25	0.09	7968
	Independent Transformer model	0.63	55.90	44.00	0.13	8435
	MLP model	0.51	64.07	48.53	0.15	8990
	LSTM model	0.55	61.13	47.07	0.15	8804
	CNN model	0.63	55.90	45.11	0.13	8595

5. Conclusions

Traditional wind power forecasting methods often ignore complex coupling relationships among different wind farms. To address this issue, this paper proposes a short-term wind power forecasting method based on multiple decoders and multi-task learning. The proposed model adopts a single-encoder–multi-decoder architecture, in which a unified encoder is used to encode the input data, while multiple decoders are employed to perform the forecasting tasks for individual wind farms, thereby accounting for the intrinsic relationships among them. Case studies based on real wind farm data from the Inner Mongolia Autonomous Region of China demonstrate that, compared with classical forecasting mod-

els, the proposed model can effectively improve forecasting performance across different wind farms. In terms of accuracy metrics, compared with the Independent Transformer, MLP, LSTM, and CNN models, the proposed model improves R^2 by up to 70.59%, reduces RMSE by up to 47.68%, reduces MAE by up to 47.97%, and reduces nRMSE by up to 40.00%. In terms of the cost metric, the proposed model reduces the daily average total operating cost by up to 11.37%, thereby lowering system operating costs and improving economic efficiency.

Despite the favorable forecasting performance, several limitations remain. First, although the multi-decoder design improves forecasting accuracy, scaling to a larger number of wind farms increases the number of decoders, which in turn raises computational and memory requirements. Second, the current evaluation is based on a single-year dataset, with the test period limited to one month, which may not fully reflect the model's generalization ability across seasons and interannual variability. Future work will focus on addressing both limitations to facilitate broader real-world deployment across different regions.

Author Contributions: Methodology, Q.L. and Y.L.; validation, X.Y.; formal analysis, S.W.; investigation, H.Z.; resources, Q.L.; data curation, X.Y.; writing—original draft preparation, Q.L., Y.L. and X.Y.; writing—review and editing, S.W. and R.L.; visualization, H.Z.; supervision, R.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science and Technology Program of Inner Mongolia Autonomous Region (Project No. 2022JBGS0044).

Data Availability Statement: The dataset presented in this study is available at https://github.com/768-lab/A-Short-Term-Wind-Power-Forecasting-Method-Based-on-Multi-Decoder-and-Multi-Task-Learning.git (accessed on 28 December 2025).

Conflicts of Interest: Authors Qiang Li, Yongzhi Liu and Siyu Wang were employed by the company Inner Mongolia Power (Group) Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Li, M.; Yang, M.; Yu, Y.; Lee, W.J. A wind speed correction method based on modified hidden Markov model for enhancing wind power forecast. *IEEE Trans. Ind. Appl.* **2021**, *58*, 656–666. [CrossRef]
2. Konstantinou, T.; Hatziargyriou, N. Regional wind power forecasting based on Bayesian feature selection. *IEEE Trans. Power Syst.* **2024**, *40*, 113–124. [CrossRef]
3. Jin, F.; Chen, X.; Yu, Y.; Li, K. An improved regularization stochastic configuration network for robust wind speed prediction. *Energies* **2025**, *18*, 6170. [CrossRef]
4. Silva, T.; Costa, A.; Vilela, O.C.; Willmersdorf, R.; dos Santos Júnior, J.V.; Alves, L.H.B.; Tyaquiçã, P.; de Lima, M.F.S.; de Souza, H.R.B.; Veleda, D. Application of the WRF model for operational wind power forecasting in northeast Brazil. *Energies* **2025**, *18*, 5731. [CrossRef]
5. Yang, M.; Huang, Y.; Wang, Z.; Wang, B.; Su, X. A framework of day-ahead wind supply power forecasting by risk scenario perception. *IEEE Trans. Sustain. Energy* **2025**, *16*, 1659–1672. [CrossRef]
6. Gokce, B.; Kayakutlu, G. Agent-based energy market modeling with machine learning and econometric forecasting for the net-zero emissions transition. *Energies* **2025**, *18*, 5655. [CrossRef]
7. Yunus, K.; Thiringer, T.; Chen, P. ARIMA-based frequency-decomposed modeling of wind speed time series. *IEEE Trans. Power Syst.* **2016**, *31*, 2546–2556. [CrossRef]
8. Shahid, F.; Zameer, A.; Muneeb, M. A novel genetic LSTM model for wind power forecast. *Energy* **2021**, *223*, 120069. [CrossRef]
9. Wan, C.; Xu, Z.; Pinson, P.; Dong, Z.Y.; Wong, K.P. Probabilistic forecasting of wind power generation using extreme learning machine. *IEEE Trans. Power Syst.* **2013**, *29*, 1033–1044. [CrossRef]
10. Wu, Q.; Guan, F.; Lv, C.; Huang, Y. Ultra-short-term multi-step wind power forecasting based on CNN-LSTM. *IET Renew. Power Gener.* **2021**, *15*, 1019–1029. [CrossRef]
11. Sarp, A.O.; Mengüç, E.C.; Peker, M.; Güvenç, B.Ç. Data-adaptive censoring for short-term wind speed predictors based on MLP, RNN, and SVM. *IEEE Syst. J.* **2022**, *16*, 3625–3634. [CrossRef]

12. Xuan, W.; Shouxiang, W.; Qianyu, Z.; Shaomin, W.; Liwei, F. A multi-energy load prediction model based on deep multi-task learning and ensemble approach for regional integrated energy systems. *Int. J. Electr. Power Energy Syst.* **2021**, *126*, 106583. [CrossRef]
13. Wang, S.; Zhang, Z. Short-term multivariate load forecasting of an integrated energy system based on a quantum weighted multi-hierarchy gated recurrent unit neural network. *Power Syst. Prot. Control* **2022**, *50*, 85–93. [CrossRef]
14. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017. Available online: https://dl.acm.org/doi/10.5555/3295222.3295349 (accessed on 28 December 2025).
15. Xiong, B.; Lou, L.; Meng, X.; Wang, X.; Ma, H.; Wang, Z. Short-term wind power forecasting based on attention mechanism and deep learning. *Electr. Power Syst. Res.* **2022**, *206*, 107776. [CrossRef]
16. Bispo Junior, D.A.; Leite, G.d.N.P.; Droguett, E.L.; de Souza, O.V.C.; Lisboa, L.A.; Cavalcanti, G.D.d.C.; Ochoa, A.A.V.; Costa, A.C.A.d.; Vilela, O.d.C.; Brennand, L.J.d.P.; et al. Short-term wind power forecasting with Transformer-based models enhanced by Time2Vec and efficient attention. *Energies* **2025**, *18*, 6162. [CrossRef]
17. Niu, Z.; Yu, Z.; Tang, W.; Wu, Q.; Reformat, M. Wind power forecasting using attention-based gated recurrent unit network. *Energy* **2020**, *196*, 117081. [CrossRef]
18. Dvorkin, V.; Delikaraoglou, S.; Morales, J.M. Setting reserve requirements to approximate the efficiency of the stochastic dispatch. *IEEE Trans. Power Syst.* **2019**, *34*, 1524–1536. [CrossRef]

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