

## Multi-temporal Scale Wind Power Forecasting Based on Lasso-CNN-LSTM-LightGBM

Qingzhong Gao\*

School of Renewable Energy, Shenyang Institute of Engineering, Shenyang 110136, China

### Abstract

Due to the increasingly severe climate problems, wind energy has received widespread attention as the most abundant energy on Earth. However, due to the uncertainty of wind energy, a large amount of wind energy is wasted, so accurate wind power prediction can greatly improve the utilization of wind energy. To increase the forecast for wind energy accuracy across a range of time scales, this paper presents a multi-time scale wind power prediction by constructing an ICEEMDAN-CNN-LSTM-LightGBM model. Initially, feature selection is performed using Lasso regression to identify the most significant variables affecting the forecast for wind energy across distinct time intervals. Subsequently, the ICEEMDAN is utilized to break down the wind power data into various scales to capture its nonlinear and non-stationary characteristics. Following this, a deep learning model based on CNN and LSTM networks is developed, with the CNN responsible for extracting spatial features from the time series data, and the LSTM designed to capture the temporal relationships. Finally, the outputs of the deep learning model are fed into the LightGBM model to leverage its superior learning capabilities for the ultimate prediction of wind power. Simulation experiments demonstrate that the proposed ICEEMDAN-CNN-LSTM-LightGBM model achieves higher accuracy in multi-time scale wind power prediction, providing more reliable decision assistance with the management and operation of wind farms.

**Keywords:** wind power prediction, ICEEMDAN, CNN network, LSTM network, LightGBM model, multi-time scale

Received on 15 November 2023, accepted on 5 March 2024, published on 15 April 2024

Copyright © Q. Gao Author *et al.*, licensed to EAI. This is an open access article distributed under the terms of the [CC BY-NC-SA 4.0](#), which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi: 10.4108/ew.5792

### 1. Introduction

As global environmental issues continue to deteriorate, an increasing number of countries are incorporating environmental protection as one of their national policies[1]. Among these issues, energy problem is a major concern for many countries, as the overindulgence in fossil fuels like coal and oil leads to significant emissions of harmful gases into the atmosphere, further deteriorating the environment. Furthermore, fossil fuels are non-renewable resources with limited reserves on Earth. As a result, an increasing number of countries are turning to renewable energy sources. Wind energy, in particular, has garnered significant attention due to its plentiful reserves on our planet. According to the World Energy Outlook, the total amount of wind power generated globally installations generation is projected to more than double by 2030, exceeding 2000 GW[2]. This indicates the increasing significance of wind energy in the global energy mix. Additionally, based on data from the International Energy Agency, wind power generation

accounted for approximately 5% of the global total electricity consumption in 2019, amounting to 1.32 trillion kilowatt-hours[3]. Accurate multi-time scale wind power forecasting can provide valuable reference for urban grid dispatching. Currently, several methods exist for wind power forecasting, including time series forecasting, single model forecasting, and ensemble model forecasting. Time series forecasting performs well when dealing with stable data, but its performance deteriorates when faced with unstable and highly fluctuating data. Ensemble models, on the other hand, can combine the strengths of different models to achieve higher forecasting accuracy and speed.. Zhang et al.[4] combined backpropagation neural networks, ARMA models, LS-SVMs and used data decomposition algorithms to decompose the original data by frequency, thereby improving wind power forecasting accuracy. Wang et al.[5] created a small-scale BP neural network that included statistical analysis and weight convergence., improved feature extraction through an improved mutual information algorithm, and significantly increased the stability, effectiveness, and anticipating wind energy precision. Farah et al.[6] used clue-based random missing (CMAR) and pattern-based K-nearest neighbor algorithm (PkNN) for feature extraction of the original data, and

\*Corresponding author. Email: gaoqz@sie.edu.cn

constructed a hybrid model using CNNs and LSTM neural networks, which effectively improved the precision of monthly forecasts for wind energy. Ning et al.[7] proposed a technique of wind energy forecasting using the EMD-CCTransformer. To begin with, the original wind power data is decomposed according to the frequency levels using the Empirical Mode Decomposition (EMD) algorithm. Subsequently, an attention mechanism from a hybrid convolutional model is integrated into the system, culminating in the formation of an entirely new model. Upon testing, this novel model has effectively enhanced the precision of the wind energy production forecast. While the aforementioned techniques have successfully increased the wind power production forecast accuracy, they can only predict a single time scale, and reprocessing of the data is required for predictions at other time scales, which has certain limitations.

In addition, temporal and spatial characteristics are also factors that cannot be ignored in wind power prediction. Zhu et al.[8] studied the problem of simultaneous wind speed prediction at multiple stations based on temporal and spatial correlation, and built a CNN-MLP wind power prediction model, which can better extract temporal and spatial characteristics of wind power to improve the prediction accuracy of the model. In addition to using the neural network model to extract the spatio-temporal characteristics, some scholars also extracted the spatio-temporal characteristics of wind power by means of graph learning. He et al.[9] used graph learning and time series analysis tools to analyze a large number of actual data of wind farms, and then used mathematical algorithms to express the probability distribution and traffic rate of wind power. And the finite state Markov chain model is used to build the prediction model of wind power, and finally the accurate prediction of wind power is realized.

To achieve the anticipated amount of wind energy produced at different time scales and further bolster the precision and speed of wind energy forecasting, this essay suggests a multi-time scale wind energy forecasting model based on Lasso-CNN-LSTM-LightGBM. It includes:

1) There are many features in the original data, and different time scales require consideration of different features. Dynamic feature extraction is performed through the Lasso model.

2) The ICEEMDAN algorithm breaks down the data frequency in ascending sequence from high to low, amplifying the potential connections between influencing factors and wind power at different frequencies.

3) A combination neural network model of CNN-LSTM-LightGBM is constructed, where CNN extracts the hidden features in the spatial properties of the data of wind power, The LSTM network extracts hidden features within the temporal characteristics of the data., and the LightGBM model further extracts and classifies the output of the plane layer. Compared to single models, this combination model has higher forecasting accuracy and speed.

## 2. Data processing

### 2.1. Feature extraction based on the Lasso

The Lasso model is a regression model based on improved linear decomposition, which achieves data classification by studying the coupling relationship between different variables[10]. The distinctive feature of the Lasso model is the introduction of an L1 regularization term in regression analysis, which allows certain coefficients to be precisely reduced to zero, thereby achieving the purpose of variable selection (feature selection). In general, regression can be expressed as follows[11]:

$$e_b = \alpha_0 + \alpha_1 \omega_{\chi_1} + \dots + \alpha_b \omega_{\chi_b} + \dots + \alpha_n \omega_{\chi_n} + \theta_{\chi} \quad (1)$$

In the given equation,  $e_b$  represents the actual value of the  $b$  th dependent variable;  $n$  represents the total count of features;  $\alpha_0$  represents the intercept;  $\alpha_b$  represents the regression coefficient of the  $b$  th feature value;  $\omega_{\chi_b}$  represents the observed value of the  $\chi$  th feature value applied to the  $b$  th feature value.  $\theta_{\chi}$  represents the residual value. The loss function can be expressed as:

$$Loss(a) = \|A - B\alpha\|^2 \quad (2)$$

In the given equation,  $Loss$  represents the loss function;  $B$  represents the dependent variable;  $A$  represents the outcome;  $\alpha$  represents the regression coefficients.

Due to the problem of overfitting that often occurs in traditional regression models, the Lasso model introduces a shrinkage parameter on top of the traditional regression model. By reducing the variance through the shrinkage parameter, the Lasso model can avoid overfitting. The model can be represented as follows:

$$L(\alpha) = Loss(\alpha) + \rho P(\alpha) \quad (3)$$

In the given equation,  $L$  represents the loss function after introducing the shrinkage parameter;  $Loss(\alpha)$  is the original loss function;  $\rho P(\alpha)$  represents the shrinkage parameter.

The Lasso model not only effectively avoids overfitting problems, but also can effectively handle situations with small data size, large number of features, and discontinuity.

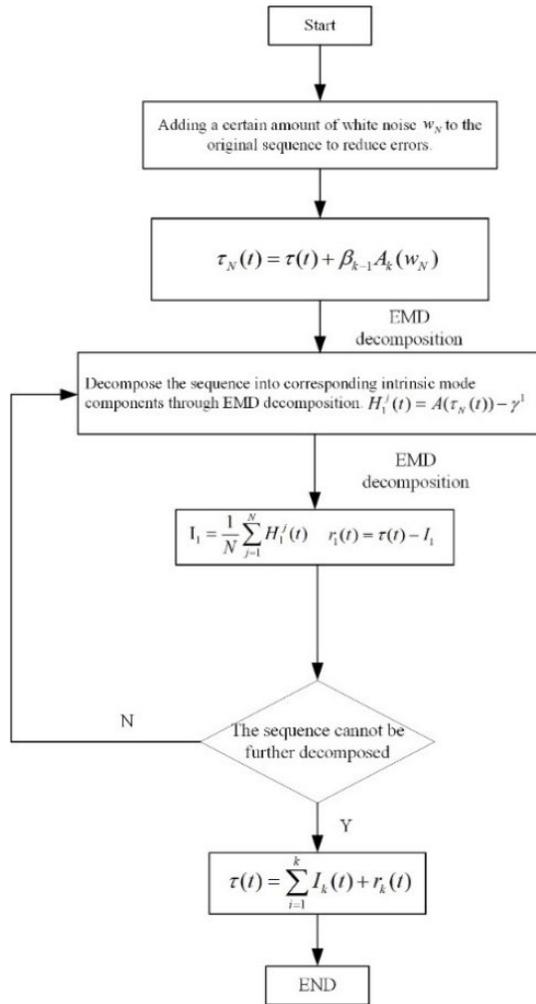
### 2.2. Processing of raw data based on ICEEMDAN

The Improved Complete Ensemble Empirical Mode Decomposition with Adaptive Noise is derived from the Empirical Mode Decomposition technique. The EMD algorithm eliminates the need for manually defining the frequency scale during the decomposition process, thereby effectively reducing errors arising from human intervention in the computational procedure[12].

However, many researchers have discovered through experimentation that when decomposing large datasets,

there can arise situations where a single data point may simultaneously represent both a maximum and a minimum value—a phenomenon known as endpoint divergence. This phenomenon of endpoint divergence can lead to significant errors, adversely affecting the predictive accuracy of the final model. Additionally, when the timescale of the data exhibits abrupt changes, intrinsic mode functions (IMFs) of different timescales may appear within the same IMF, stripping the component of any practical physical significance and affecting the final prediction accuracy.

Figure 1 displays the procedure diagram for data decomposition using ICEEMDAN.



**Figure 1.** The procedure diagram for data decomposition using ICEEMDAN

ICEEMDAN effectively addresses the endpoint effect and mode mixing problems encountered in the EMD decomposition process. The computational steps involve:

1) Determine the number of times  $N$  to add noise and obtain  $N$  initial quantities  $\tau_N(t)$ :

$$\tau_N(t) = \tau(t) + \beta_{k-1} A_k(w_N) \quad (4)$$

In the given equation,  $A_k$  is the  $k$ th component acquired through EMD,  $\gamma^j$  represents residuals acquired through EMD,  $\beta$  represents a constant.  $w$  is a noise that does not affect the sequence.

2) Take the average of all  $I_1$  values to obtain the first-order IMF component from the ICEEMDAN decomposition as well as the residual  $r_1(t)$  [13]:

$$I_1 = \frac{1}{N} \sum_{j=1}^N A(\tau_N(t)) - \gamma^1 \quad (5)$$

$$r_1(t) = \tau(t) - I_1 \quad (6)$$

3) Add Gaussian white noise to  $r_1(t)$  to obtain the second-order data, and the decomposed sequence is  $H_1^j(t)$  [14]:

$$H_1^j(t) = A(r_1(t) + \beta_{k-1} A_k(w_N)) - \gamma^2 \quad (7)$$

4) Take the average of  $H_1^j(t)$  to acquire the IMF2 in ICEEMDAN decomposition:

$$I_2 = \frac{1}{N} \sum_{j=1}^N H_1^j(t) \quad (8)$$

5) Perform the aforementioned procedure until the raw data  $r_k(t)$  can no longer be decomposed. The number of IMF is  $k$ . The raw data is decomposed as follows:

$$\tau(t) = \sum_{i=1}^k I_i(t) + r_k(t) \quad (9)$$

## 3. Model building

### 3.1. Convolutional Neural Networks

Convolutional Neural Networks have been structurally optimized compared to traditional neural networks and It is mainly divided into five functional parts: input, convolutional, pooling, fully connected, and output[15]. Data from the input layer is sent into the convolutional layer and performs local extraction before passing it on to a nonlinear activation function, which calculates the output of each neuron within the convolutional layer. Within the same output plane, CNNs maintain a weight sharing computation strategy, which has the ability to speed up a neural network's training process and decrease model complexity. After a convolutional neural network, its result is:

$$N_j^{out} = \phi(N_1^{in} \omega_1 + N_2^{in} \omega_2 + \dots + N_k^{in} \omega_k + b_n) \quad (10)$$

In the given equation,  $N_j^{out}$  represents the output value of a neuron,  $N_1^{in}$  stands for a neuron's input value,  $\omega_k$  stands for the weights,  $\phi$  represents the activation function, and  $b_n$  represents the bias term[16].

After a convolutional layer, a nonlinear activation function is typically employed. The purpose of this function is to introduce nonlinearity to tackle nonlinear problems and

enhance the model's ability to process complex data. Activation functions apply a nonlinear transformation to feature values, thereby boosting the network's expressive power.

Pooling layers, also known as subsampling layers, primarily serve to decrease the feature maps' spatial dimensionality, cutting down on the amount of computation and the number of parameters.

Fully connected layers usually follow the convolutional and pooling layers. Their role is to integrate the learned high-level features for tasks such as classification or other

functions. The fully connected layer connects the neurons between the previous and the former to enable the correlation of data. These layers often utilize the Softmax function to compute probability distributions or classify data.

Normalization layers, which include local response normalization or batch normalization, help to enhance the network's generalization capabilities, speed up training, and reduce dependence on the initialization of weights. Normalization layers standardize the feature maps, ensuring a more stable training process for the network. Figure 2 shows computational process of the CNN.

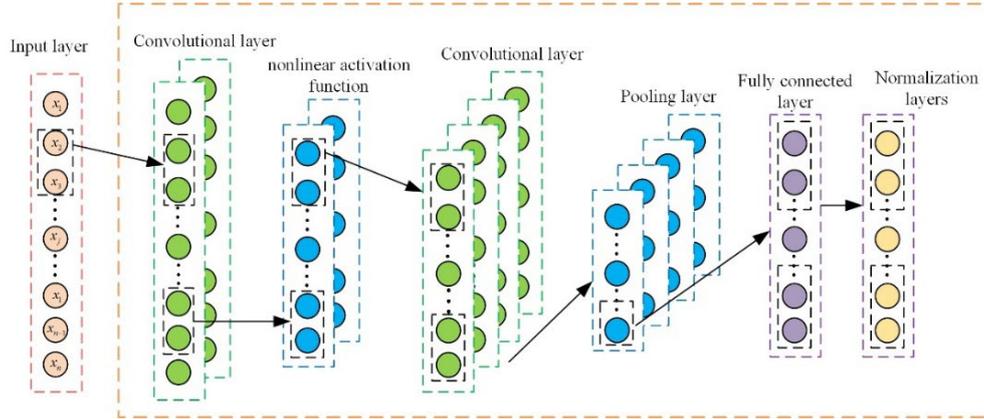


Figure 2. The computational process of the CNN

### 3.2. LSTM Neural Networks

An enhanced variant of RNNs are Long Short-Term Memory networks. LSTM is designed to address the difficulties traditional RNNs face when processing long-term dependencies. In contrast to the conventional recurrent neural network models, LSTMs incorporate four key components: input, output, forget, and memory components. The role of the forget gate is to selectively discard information during the neural network training process to ensure better data propagation, while the memory cell is tasked with retaining information. This kind of operation process effectively solves various problems in the process of recurrent network operation, and greatly improves the accuracy of prediction[17].

#### (1) Forget Gate

The forget gate determines what type of information the neurons in the LSTM should discard, using the Sigmoid function to facilitate the forgetting of information from the previous step's neurons. The operation formula is as follows:

$$forget_{\rho} = \mu(\omega_{forget}[U_{\rho}, x_{\rho}] + b_{\rho}) \quad (11)$$

In the given equation,  $\mu$  represents the Sigmoid activation function,  $\omega_{forget}$  represents the weight,  $U_{\rho}$  represents the output value of the preceding neuron,  $x_{\rho}$  reflects the neuron's input value., and  $b_{\rho}$  stands for the bias correction term[18].

#### (2) Input Gate

In a Long Short-Term Memory (LSTM) network, the input gate selects the information that the neurons should retain and enters it into the neuron's state. The input gate consists of two parts: the Sigmoid gate and the tanh gate. The Sigmoid gate determines which components of the neuron to update, while the tanh gate is responsible for creating an alternate component. Finally, these two components are combined together. The update component for the Sigmoid gate is shown in Equation 12, and the update component for the tanh gate is shown in Equation 13.

$$Input_{\rho} = \mu(\omega_{\rho}[U_{\rho}, x_{\rho}] + b_{\rho}) \quad (12)$$

$$T_{\rho} = \tanh(i_{\rho}\omega_{\rho}[U_{\rho}, x_{\rho}] + b_{\rho}) \quad (13)$$

In the given equation,  $\omega_{\rho}$  A represents the weights of the input gate.

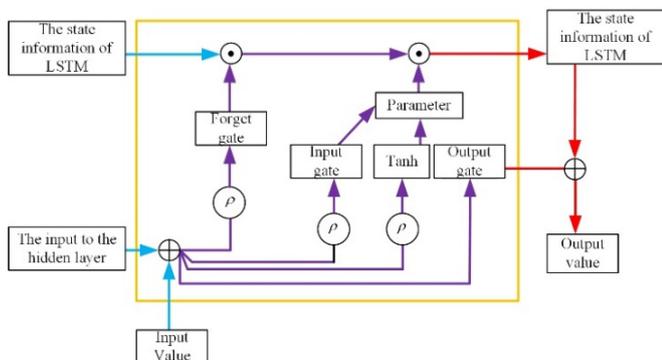
#### (3) Output Gate

The main capability of the output gate is to output the data predicted by the network. After receiving the information passed on from the previous neuron, it computes the output value using the Sigmoid function. Here is the formula:

$$Out_{\rho} = \mu(\omega_o[U_{\rho}, x_{\rho}] + b_o) \quad (14)$$

The operation formula is as follows:  $Out_{\rho}$  represents the final output of information.  $\mu$  represents the Sigmoid activation function,  $\omega_o$  represents the weight,  $U_{\rho}$  represents

the output value of the preceding neuron,  $x_p$  reflects the neuron's input value, and  $b_o$  stands for the bias correction term. Figure 3 shows the structural diagram of the LSTM.



**Figure 3.** The structure of the LSTM

### 3.3. LightGBM regression model

LightGBM is a decision tree algorithm framework developed by Microsoft Corporation, which aims to improve the overall computing speed of the model. Through experiments, it is found that the LightGBM model has excellent ability to process a large amount of data, and its core distributed computing method is to first divide the data into multiple parts, and then perform gradient operation on each part to finally realize the model's capacity to make precise predictions.

LightGBM is characterized by a distributed decision tree algorithm, which can significantly reduce the complexity of the computation process and simplify the computation process in the recalculation process. In particular, it has good ability to deal with discrete data that are free from continuous characteristic curves. In addition, LightGBM has two special structures compared to other models[19]:

1. Gradient-based One-Side Sampling: This structure is a special structure for collecting data, which allows the LightGBM model to exclude the data with a modest gradient during the recalculation phase and keep only the data with a big gradient. This structure allows it to reduce the amount of data that needs to be processed without ensuring the overall continuity of the data.

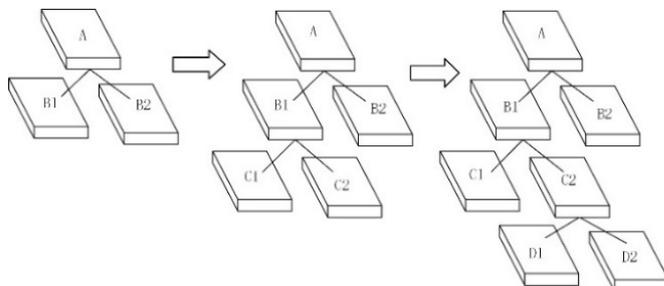
2. Exclusive Feature Bundling (EFB): EFB is a feature dimensionality reduction technique that bundles mutually exclusive features into a single feature. By bundling sparse features, EFB reduces the number of features that need to be processed, thus decreasing computational complexity.

LightGBM also supports various parallel learning strategies and can handle a large number of categorical features without the need for explicit feature transformation. LightGBM continuously divides the original data into calculations by establishing a decision tree, and the prediction results are expressed as[20]:

$$F_{\kappa}(x) = \sum_{\beta=1}^{\kappa} F_{\beta}(x) \quad (15)$$

In the given equation,  $F_{\kappa}(x)$  represents the anticipated amount;  $F_{\beta}(x)$  is the resultant amount of the  $\beta$  th tree.

Figure 4 shows the data splitting process of LightGBM.



**Figure 4.** The data splitting process of LightGBM

### 3.4. CNN-LSTM-LightGBM model

This part mainly introduces the CNN-LSTM-LightGBM neural network combination model, which integrates the advantages of the three models and compensates for the shortcomings of each other. The model can realize wind power prediction at different time scales. The following is the model's prediction process:

After determining the time scale of wind power prediction, Lasso model is used to extract the wind power characteristics under the changed time scale. The second step employs the original time series into many Intrinsic Mode Function components using the ICEEMDAN decomposition technique. The third step involves feeding the information into the CNN module to extract latent feature information contained within the data. The fourth step utilizes the LSTM module to extract time-sequential features from the data. The fifth step involves optimizing the forecasted information through the LightGBM module, culminating in the output of data that satisfies the convergence criteria. Figure 5 shows process diagram of CNN-LSTM-LightGBM.

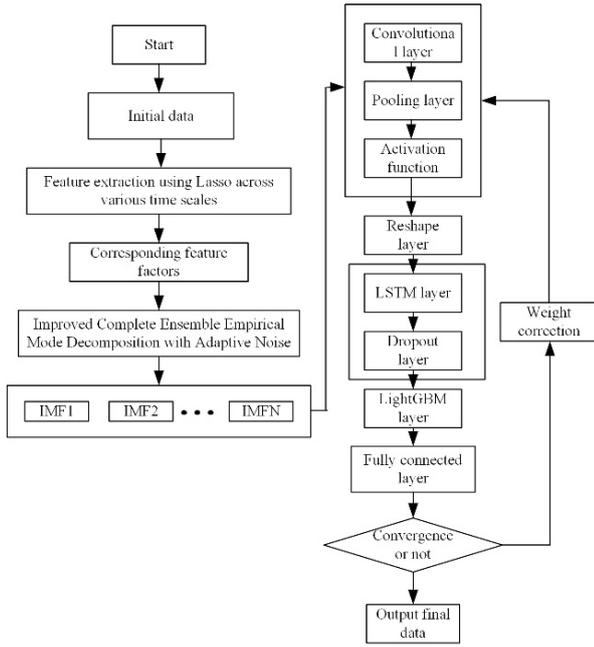


Figure 5. Process diagram of CNN-LSTM-LightGBM

### 4. Model simulation

Using actual operational information from a wind power plant in Liaoning, China, with a wind power data sampling interval of 5 minutes, this study focuses on ultra-short-term and short-term wind power forecasting. The Lasso-ICEEMDAN-CNN-LSTM model was implemented using the Matlab2020a programming language on a computing platform running the Windows 10 operating system, with an Intel i7-12700 processor, RTX3060Ti graphics card, and 32GB of RAM.

Firstly, the wind power dataset including seven feature vectors is input into the Lasso model. The Lasso model extracts features based on the actual forecasting temporal scale requirements, selecting those with the greatest impact. Subsequently, the ICEEMDAN model decomposes the features and wind power data. The decomposed dataset is then fed into the CNN-LSTM-LightGBM model. Within this model, the CNN component receives input data of size 1x6, with a convolution kernel of 3 and a pooling layer of size 1x1. After convolution, pooling, and fully connected computations, the processed feature information is flattened and used as input for the LSTM model. The LSTM model's neuron size and number of hidden layers have an impact on the predicting inaccuracy and require optimization to achieve the best network structure. Testing revealed that the first layer of the LSTM neural network has 64 neurons, and the second layer has 32 neurons. The LightGBM model has a maximum tree depth of 3, a count of 50 learners, a learning rate of 0.01, a complexity control of 32, a booster type of gbdt, and a minimum record count of 25 for the leaves[21].

To verify the superiority of the Lasso-CNN-LSTM-LightGBM model, this paper selects three other models for comparative analysis: the BP neural network, and SVM, across both very short-term and short-term forecasting time scales. Figure 6 shows the wind speed after decomposition by ICEEMDAN.

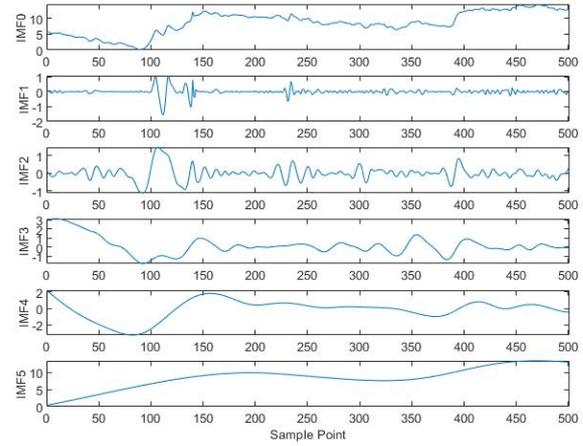


Figure 6. The wind speed after decomposition by ICEEMDAN

Figure 7 shows the wind power after decomposition by ICEEMDAN, and Figure 8 shows the predicted values for each IMF. In Figure 8, it can be observed that the errors primarily manifest in the IMF component with the highest frequency. As illustrated in Figure 9, the comparison of predicted values versus actual values for three different models is presented. It is evident from the figure that the CNN-LSTM-LightGBM model achieves the highest predictive accuracy. In this study, statistical assessment metrics such as the symmetric Mean Relative Error (s\_MRE) and the Mean Absolute Error (MAE) are utilized as evaluation criteria. The error metrics for the predicted values of each model are presented in Table 1.

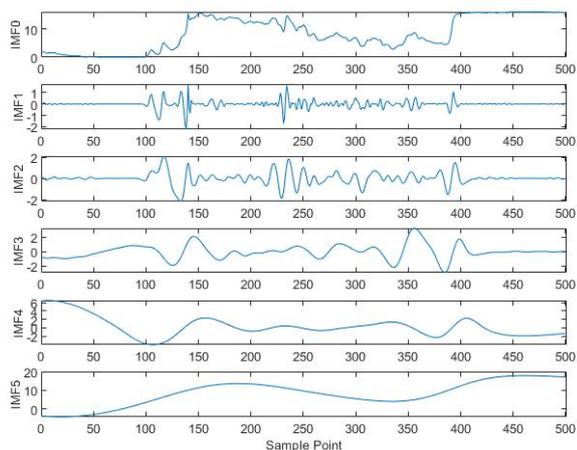


Figure 7. The wind power after decomposition by ICEEMDAN

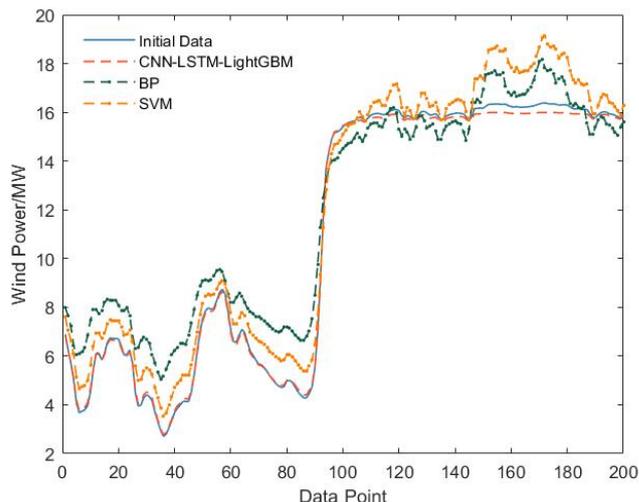


Figure 9. The comparison of predicted values versus actual values for three different models

Table 1. The error metrics for the predicted values of three models

Model	s_MRE	MAE
CNN-LSTM-LIGHTGBM	86.8%	0.037
BP	60.02%	0.091
SVM	47.25%	0.169

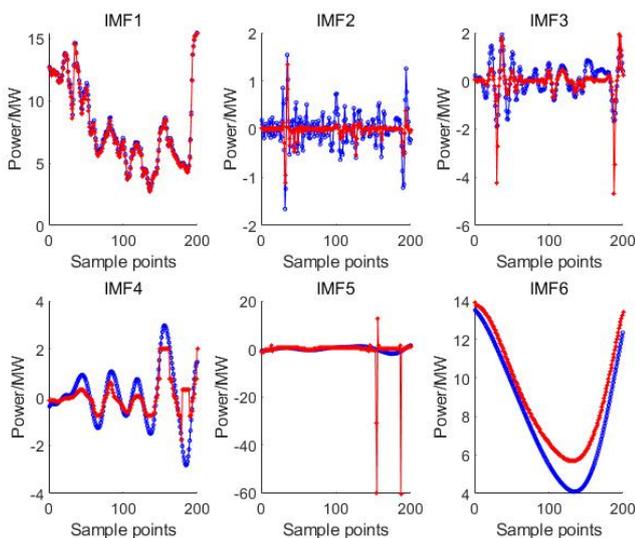


Figure 8. The predicted values for each IMF

### 5. Conclusion

An approach to multi-time scale wind power forecasting is proposed in this essay. It combines the ICEEMDAN algorithm with the Lasso-CNN-LSTM-LightGBM composite model. Using actual data from a wind power plant in Liaoning for simulation analysis, the proposed composite model demonstrates superior forecasting performance when compared to the BP and SVM models.

### 6. Discussion

The limitation of this paper is that at present, most of the data of wind power influencing factors are collected from the NWP database, so the prediction accuracy of wind power is largely limited by the data accuracy provided by the NWP database. After ICEEMDAN algorithm is used to decompose wind power data, the prediction accuracy of the residual component of the neural network model is low. In the future, a model can be built to solve the above problems to improve the original data, and the residual component can be improved separately to improve the prediction accuracy of the model.

## Acknowledgement

This work was supported by Shenyang Science and Technology Plan Project (22-322-3-29) .

## References

- [1] Yuxin Gong, Bo Wang, Huaizheng Ren, Deyu Li, Dianlong Wang, Huakun Liu and Shixue Dou. Recent Advances in Structural Optimization and Surface Modification on Current Collectors for High-Performance Zinc Anode: Principles, Strategies, and Challenges. *Nano-Micro Letters*. 2023; 15(11):325-356.
- [2] Xixian G. Energy Transition:Huge Potential For China-ASEAN Cooperation. *China Report ASEAN*. 2023;8(Z2):26-29.
- [3] Yachen Xie, Xuning Wu, Zhengmeng Hou, Zaoyuan Li, Jiashun Luo, Christian Truitt Lüddecke, Liangchao Huang, Lin Wu and Jianxing Liao. Gleaning insights from German energy transition and large-scale underground energy storage for China's carbon neutrality. *International Journal of Mining Science and Technology*. 2023; 33(05):529-553.
- [4] ZHANG, G., LIU, H. and ZHANG, J. Wind power prediction based on variational mode decomposition multi-frequency combinations. *J. Mod. Power Syst. Clean Energy*. 2019; 7:281–288.
- [5] Wang, SX., Li, M. and Zhao, L. Short-term wind power prediction based on improved small-world neural network. *Neural Comput & Applic*. 2019; 31:3173–3185.
- [6] Farah S, Atif M and Rizwan K. Short-term wind power prediction based on improved small-world neural network. *Neural Comput & Applic*. 2019; 31:3173–3185.
- [7] Ning L, Jie D and Lingyue L. A novel EMD and causal convolutional network integrated with Transformer for ultra short-term wind power forecasting. *International Journal of Electrical Power and Energy Systems*. 2023;154.
- [8] Zhu Q, Chen J and Zhu L. Wind speed prediction with spatio-temporal correlation: A deep learning approach. *Energies*, 2018, 11(4):705.
- [9] He M, Yang L and Zhang J. Spatio-temporal analysis for smart grids with wind generation integration. 2013 International Conference on Computing, Networking and Communications (ICNC). 2013:1107-1111.
- [10] Jiaqi Liang, Chaoye Wang, Di Zhang, Yubin Xie, Yanru Zeng, Tianqin Li, Zhixiang Zuo, Jian Ren and Qi Zhao. VSOLassoBag:a variable-selection oriented LASSO bagging algorithm for biomarker discovery in omic-based translational research. *Journal of Genetics and Genomics*. 2023; 50(03):151-162.
- [11] Meihong S and Wenjian W . A network Lasso model for regression. *Communications in Statistics-Theory and Methods*. 2023; 52(6):1702-1727.
- [12] Saibabu B B, Kiran T and Vishaljeja K . Hybrid wind speed forecasting using ICEEMDAN and transformer model with novel loss function. *Energy*. 2023; 265
- [13] Ning Li, Jie Dong, Lingyue Liu, He Li and Jie Yan. A novel EMD and causal convolutional network integrated with Transformer for ultra short-term wind power forecasting. *International Journal of Electrical Power & Energy Systems*, 2023; 154:109470.
- [14] Chenjia Hu, Yan Zhao, He Jiang, Mingkun Jiang, Fucai You, Qian Liu. Prediction of ultra-short-term wind power based on CEEMDAN-LSTM-TCN. *Energy Reports*. 2022; 8:483-492.
- [15] Matin S M, Rikhtehgar A G and Aminzadeh A G. Predicting wind power generation using machine learning and CNN-LSTM approaches. *Wind Engineering*. 2022; 46(6):1853-1869.
- [16] Anggraini Puspita Sari, Hiroshi Suzuki, Takahiro Kitajima, Takashi Yasuno, Dwi Arman Prasetya and Rahman Arifuddin. Short-Term Wind Speed and Direction Forecasting by 3DCNN and Deep Convolutional LSTM. *IEEJ Transactions on Electrical and Electronic Engineering*. 2022; 17(11):1620-1628.
- [17] Ahmed A. Ewees, Mohammed A.A. Al-qaness, Laith Abualigah and Mohamed Abd Elaziz. HBO-LSTM: Optimized long short term memory with heap-based optimizer for wind power forecasting. *Energy Conversion and Management*. 2022; 268
- [18] Indolia S, Goswami A K and Mishra S P. Conceptual understanding of convolutional neural network-a deep learning approach. *Procedia computer science*. 2018; 132:679-688.
- [19] Shijie G, Yongsheng W and Limin L.Ultra-short-term wind power prediction method combining financial technology feature engineering and XGBoost algorithm. *Heliyon*. 2023; 9(6):e16938-e16938.
- [20] Ju Y, Sun G and Chen Q. A Model Combining Convolutional Neural Network and LightGBM Algorithm for Ultra-Short-Term Wind Power Forecasting. *IEEE Access*. 2019; 7:28309-28318.
- [21] Shengli L, Xudong T and Benxi L. Short-Term Wind Power Prediction Based on LightGBM and Meteorological Reanalysis. *Energies*. 2022; 15(17):6287-6287.