Periodic-Enhanced Informer Model for Short-Term Wind Power Forecasting Using SCADA Data

Zhao-Hua Liu, *Senior Member*, *IEEE*, Long-Wei Li, Hua-Liang Wei, Ming Li, Ming-Yang Lv, and Ying-jie Zhang

***Abstract*—Supervisory Control and Data Acquisition (SCADA) systems can collect abundant information about wind farm operation and environment. To better make use of SCADA data, a periodic-enhanced informer model for short-term wind power forecasting using scada data is proposed. Firstly, to effectively filter out noise in SCADA data, a *v-p* curve-based method is adopted by incorporating a quartile approach to remove sparse outliers; a density-based spatial clustering of applications with noise (DBSCAN) algorithm is then employed to eliminate stacked outliers from the power curve. Secondly, a multi-feature input set selection method based on Maximization Information Coefficient is introduced to make better use of the SCADA system data by reducing the number of features. Thirdly, a Temporal Convolutional Network (TCN) is designed to extract the scalar projection of the input set, followed by fusing the local time stamp and global time stamp to build the periodic information enhanced prediction model embedding layer. Subsequently, the enhanced input set is fed into an informer model to predict future wind power. Finally, considering the multiple temporal scales structure characteristics present in the dataset, a multi-scale deep fusion module is established in the informer model to deeply integrate the features of different scales. It can simultaneously avoid the resource waste and overfitting problems caused by increasing the network depth. The experimental results, which are obtained from several deep learning methods on real SCADA data, demonstrate that the suggested approach has good predictive capability.**

***Index Terms*—SCADA, wind power forecasting, TCN, informer, maximum information coefficient.**

I. Introduction

In the face of the gradual depletion of fossil fuels, countries worldwide are shifting their focus towards clean and renewable energy as a pivotal avenue for development [1]. Among the most promising renewables, wind energy has surfaced as a key factor to the transition to sustainable energy, leveraging advantages such as abundant resources, less pollution emissions, and high-cost effectiveness [2]. To this end, wind farms have implemented a range of measures to effectively utilize wind energy, such as enhancing grid stability, refining distribution plans, and optimizing power generation schedules [3]. Among these measures, short-term wind power forecasting (WPF) stands as a critical component, providing decision-making foundations for power system dispatchers [4].

WPF can generally be classified into two categories: methods based on physics and data-driven [5]. The physical technique, such as numerical weather prediction, which utilizes inputs like wind speed, wind direction, and other meteorological data to simulate the trajectory of wind for predicting wind power [6]. Nevertheless, the application of physical models is severely limited by atmospheric conditions and computational complexity [7].

With the increasing prevalence of SCADA systems in wind turbines, a substantial volume of data has been collected. Consequently, data-driven methodologies are becoming the mainstream choice for solving the challenge of wind power prediction. The typical data-driven methods primarily consist of statistical methods and artificial intelligence methods. The traditional parametric statistical methods mainly include autoregressive model [8], autoregressive moving average model [9], and autoregressive integrated moving average (ARIMA) model [10]. These methods can only establish linear relationships among data, but cannot effectively represent nonlinear dynamics of the related processes [11].

Conversely, artificial intelligence methods have gradually taken the forefront. In [12], a heteroskedastic support vector regression (SVR) model, distinguished from the conventional SVR, was introduced to effectively learn the uncertainty inherent in the sequence. In [13], a combination model based on feature selection, utilizing a convolutional neural network (CNN) and bidirectional long and short-term memory network (LSTM) was proposed to achieve good results on the KDD Cup 2022 dataset. In [14], an approach based on LSTM combined with entropy and mutual information (MI) features selection techniques achieved satisfactory wind power prediction results. In [15], Zhu *et al.* proposed a TCN model. This technique is specifically designed for time-series forecasting and has been applied to other prediction tasks. In [16], an attention TCN based on stacked extended causal convolution and attention mechanism was proposed for the point and probabilistic forecasting of renewable resources. In [17], the TCN was employed to predict the components of the decomposition-reconstruction and finally integrated them to obtain wind power predictions. The above methods are adept

Manuscript received July 13, 2024, first revised October 12, 2024, second revised January 23, 2025, and accepted March 31, 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62473147, and in part by the Hunan Provincial Key Research and Development Project of China under Grant 2022WK2006, and in part by the International Exchanges 2022 Cost Share Programme Between the Royal Society and the National Natural Science Foundation of China (NSFC) under Grant IEC\NSFC\223266.

Z.-H. Liu, L.-W Li, M. Li and M.-Y. Lv are with the School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan 411201, China (e-mail: zhaohua.liu@hnust.edu.cn; lilongwei0316@163.com; minglee@hnust.edu.cn; 1040133@hnust.edu.cn).

H.-L. Wei is with the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield S1 3JD, U.K. (e-mail: w.hualiang@sheffield.ac.uk).

Y.-J Zhang is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: zhangyj@hnu.edu.cn).

at learning potential nonlinear relationships from data, exhibiting higher flexibility and accuracy [18]. The power output of wind turbines is not solely contingent upon environmental factors, it also depends on the historical previous status of the turbines [19]. The effective utilization of SCADA systems, which accumulate a substantial amount of data on environmental and turbine operation, is of critical importance in improving the accuracy of WPF [20]. In [21], the SCADA data was subjected to analysis in three dimensions, namely spatial, physical, and temporal features, employing a diverse range of techniques, including integrated K-means, K-shape, CNN, and gated recurrent units (GRU). In [22], Liu *et al.* presented a deep learning framework for WPF using SCADA data that utilizes wavelet decomposition-based denoising followed by LSTM networks for training. In [23], the isolated forest algorithm was employed to detect anomalies in SCADA data, and then a deep learning network was constructed to map the characteristic graph to wind power prediction.

Short-term forecasting can usually be achieved through either a recursive or a direct approach, together with a multi-input multi-output strategy [24] to address the multi-step prediction problem, typically limited to predicting only 48 data points [25]. For longer prediction data points, these methods face an accumulation of computational burden and can usually focus only on the wind power information contained in the data points, while overlooking the internal connections between the data points [26]. Consequently, it would be necessary to further explore advanced models or novel inference architectures to improve the precision of short-term WPF.

TABLE I
THE MEMORY AND TIME COMPLEXITY OF SOME MODELS

Parameters	Autoformer	Reformer	Pyraformer	Crossformer
Memory	$O(L\log L)$	$O(L\log L)$	$O(L)$	$O(L)$
Time	$O(L\log L)$	$O(L\log L)$	$O(L)$	$O(DT^2/L_{seg}^2)$

Inspired by the widespread success of Transformer [27] in computer vision, some scholars have endeavored to apply it to time-series forecasting. However, owing to the memory and time complexity of Transformer being $O(L^2)$, where L denotes the length of the input time series, it is impractical to apply Transformer directly to WPF [28]. Hence, attempts have been made to propose suggestions for improvement. In [29], a Sparse Transformer based on two modes strided and fixed was introduced to achieve sparsity in the Transformer's self-attention matrix. In [30], a decomposition framework was proposed to decompose sequences into periodic and trending components, which were embedded in an encoder-decoder structure. Moreover, an auto-correlation technique was implemented to facilitate serial connectivity. In [31], a model called Pyraformer based on the pyramid attention module was proposed to reduce spatial and temporal complexity and establish the long range dependence of time-series. Despite these models have successfully reduced the complexity of the self-attention mechanism, their efficiency improvement is modest [32]. In [33], Reformer was proposed to enhance model efficiency and scalability by employing local sensitive hashing and reversible residual layers. However, it is worth noting that hash functions may lead to information loss, and the use of reversible residual layers might impact gradient propagation. Table I shows the memory and time complexity of some models. All these methods primarily focus on optimizing the high-level complexity of the Transformer through the self-attention [34], but pay little attention to the issues of memory bottleneck when stacking layers, and a substantial reduction in speed when predicting long outputs.

In order to address these limitations, a model named Informer [35] was proposed to decrease the quadratic time complexity of the Transformer based on ProbSparse self-attention. Simultaneously, to address the stacking layers problem with long inputs, distillation layers were introduced, significantly reducing overall space complexity. To tackle the issue of speed reduction in predicting long outputs, a generative decode was proposed to acquire the complete long outputs with a single forward step. Therefore, the informer model is more suitable for WPF. However, it needs to be kept in mind the following challenges in WPF.

1) Wind power data is distinctly periodic, with peaks usually occurring in the afternoon to early evening [36]. Consequently, the extraction and effective integration of this periodic information into the Informer model represent a matter of paramount importance.
2) WPF based on SCADA data involves numerous monitoring points of wind turbines and intricate coupling relationships among monitoring parameters. However, the informer model lacks the ability to construct a multi-feature input set strongly correlated with wind power.
3) The single time scale historical data in the original SCADA system data contains limited feature quantity, which cannot fully reflect the inherent dynamics of the time-series. The question of how to extract features on multiple time scales and fuse them to obtain more comprehensive information is worthy of consideration.

To tackle the aforementioned challenges, a periodic-enhanced informer model for short-term WPF using SCADA data is proposed. Firstly, a *v-p* curve-based method is adopted, where a quartile method is utilized to remove sparse outliers and DBSCAN algorithm is employed to eliminate stacked outliers from the power curve. Secondly, features that are highly correlated with wind power are selected by applying the maximal information coefficient (MIC). Thirdly, a novel periodic enhanced information embedding layer is constructed to improve the model's capability to perceive long sequences. Finally, an informer model is designed by introducing a multi-scale deep fusion module to achieve more precise wind power prediction. The model aims to mitigate error accumulation and considers the correlation between multi-step prediction tasks, effectively capturing the permanent dependence between input and output variables. Its primary contributions are enumerated as follows:

1) A multi-feature input set is constructed. The outliers in the set are eliminated using the *v-p* curve, and features are selected based on their correlation strength with wind power, computed through the MIC theory approach.

2) A novel periodic information enhanced embedding layer is proposed. The TCN is employed to extract the scalar projection of the input set, followed by fusing the local time stamp and global time stamp.
3) A multi-scale deep fusion module is designed. The purpose is to effectively fuse local multi-scale and global characteristics to expand the width of the informer, thereby enhancing the perception of dynamic sequences.

The problem formulation and the fundamental mechanism are delineated in Section II. Section III offers a thorough elaboration of the suggested methodological framework. Experimental data is utilized in Section IV to validate the suggested approach efficacy. The conclusions are concisely summarized in Section V.

II. Preliminaries

A. Problem formulation

Assuming that a SCADA system data is composed with m samples, where each sample contains n relevant features, together with wind power value. Denote by X the dataset of interest, which is defined as follows:

$$X=\begin{bmatrix} X_1 \\ X_2 \\ \cdots \\ X_m \end{bmatrix}=\begin{bmatrix} x_{11},x_{12},\cdots,x_{1n},y_1 \\ x_{21},x_{22},\cdots,x_{2n},y_2 \\ \cdots \\ x_{m1},x_{m2},\cdots,x_{mn},y_m \end{bmatrix} \tag{1}$$

where X_m represents the m-th sample, x_{mn} and y_m denote the n-th relevant feature value of m-th sample and the wind power value of m-th sample.

The wind power values for the next $t + r$ time instant are predicted utilizing the t past samples. The input and output data are specified as follows:

$$INPUT=\begin{bmatrix} IX_1 \\ IX_2 \\ \cdots \\ IX_{m-t-r+1} \end{bmatrix}=\begin{bmatrix} X_1,X_2,\cdots,X_t \\ X_2,X_3,\cdots,X_{t+1} \\ \cdots \\ X_{m-t-r+1},X_{m-t-r+2},\cdots,X_{m-r} \end{bmatrix} \tag{2}$$

$$OUTPUT=\begin{bmatrix} OY_1 \\ OY_2 \\ \cdots \\ OY_{m-t-r+1} \end{bmatrix}=\begin{bmatrix} P_{t+1},P_{t+2},\cdots,P_{t+r} \\ P_{t+2},P_{t+3},\cdots,P_{t+r+1} \\ \cdots \\ P_{m-t-r+1},P_{m-t-r+2},\cdots,P_m \end{bmatrix} \tag{3}$$

In accordance with (2) and (3), the WPF assignment is designated as a single-step prediction when $r = 1$. Conversely, the WPF assignment is a multi-step prediction when $r \geq 2$.

B. An abnormal data cleaning algorithm for wind turbine power curve

In the operational environment of wind farms, numerous outliers are frequently recorded by SCADA data, mainly due to significant uncertainties. Data cleaning, which is focused on improving data quality, is recognized as a crucial step in mining operational data from wind farms [37]. In this paper, a *v-p* curve-based method is adopted by incorporating a quartile approach to remove sparse outliers. The DBSCAN algorithm is then employed to eliminate stacked outliers from the power curve.

The Quartile method for eliminating sparse outliers: Initially, the quartile method is applied twice. The first analysis is for the wind power observations falling within a wind speed range, whereas the second analysis concentrates on the wind speed observations corresponding to a wind power range. Taking the former as an example, the quartiles for the sequence $P_v=\{p_1,p_2,\cdots,p_s\}$, consisting of s ascendingly ordered wind power observations are calculated as follows (s represents the total number of values contained in the P_v.) :

Calculate the second quartile P_2.

$$P_2=\begin{cases} p_{\frac{n+1}{2}} & n=2k+1;k=0,1,2,\ldots \\ \dfrac{p_{\frac{n}{2}}+p_{\frac{n}{2}+1}}{2} & n=2k;k=1,2,\ldots \end{cases} \tag{4}$$

Calculate the first quartile P_1 and third quartile P_3.

When n=2k (k = 1, 2, 3, ···), P_v is split into two parts by P_2 point. The second quartile P_2' and P_2" (P_2' < P_2") of the two parts are calculated by Eq. (4), and $P_1 = P_2$', $P_3 = P_2$".

When n = 4k+1 (k = 1, 2, 3, ···),

$$\begin{cases} P_1=0.25p_k+0.75p_{k+1} \\ P_3=0.75p_{3k+1}+0.25p_{3k+2} \end{cases} \tag{5}$$

When n = 4k+3 (k = 1, 2, 3, ···),

$$\begin{cases} P_1=0.75p_{k+1}+0.25p_{k+2} \\ P_3=0.25p_{3k+2}+0.75p_{3k+3} \end{cases} \tag{6}$$

The interquartile range (IQR) can be obtained from the two quartiles as:

$$IQR=P_3-P_1 \tag{7}$$

Based on IQR, the outlier inner limit for the sequence P_v can be determined as:

$$[F_l,F_u]=[P_1-1.5IQR,P_3+1.5IQR] \tag{8}$$

All data beyond the inner limit [F$_l$, F$_u$] are outliers.

The quartile method provides a simple and efficient approach for detecting outliers in a dataset. Nevertheless, its application is confined to scenarios where outliers are present in small quantities. When confronted with the SCADA data containing a considerable number of stacked outliers, this method may become ineffective.

The DBSCAN algorithm for eliminating stacked outliers: The advantage of DBASCA lies in its ability to automatically determine cluster divisions, based on the neighborhood radius R and density threshold ε, without determining the number of clusters. For the sequence P_v, the basic concept of the DBSCAN method is defined as follows.

***R* neighborhood**: The R neighborhood of p_i is defined as $N_\varepsilon(p_i)$.

$$N_\varepsilon(p_i)=\{p_j\in P_v \mid d(p_i,p_j)\leq\varepsilon\} \tag{9}$$

where $d(p_i, p_j)$ denotes the calculation of the distance function between p_i and p_j.

Core point: A point p_i is said to be a core point if the number of data points within its neighborhood is equal to or exceeds a specified threshold ε.

Directly density-reachable: For a core point p_i, if p_j is

located within the R neighborhood of p_i, it can be asserted that p_j is directly density-reachable from p_i.

Density-reachable: If a chain of points p_i, p_{i+1}, ⋯, p_j exists such that each subsequent point p_{i+1} is directly density-reachable from the preceding point p_i, then the point p_j is considered to be density-reachable from the point p_i.

Density-connected: If two points p_i and p_j are density-reachable from a point $a \in P_v$, then p_i and p_j are said to be density-connected.

The processing flow for a set of data points in the DBSCAN algorithm is outlined as follows:

1) Set the values of ε and R.

2) All core points are identified and extracted from set P_v, thereby forming the core point set Ψ. The remaining points are then labelled as border points or noise points. In the event that the core point set Ψ is found to be empty, this signifies the termination of the clustering algorithm.

3) For each core point p_i in the set Ψ, the set of core points connected by density-reachable forms a cluster. These clusters together form a set called Ω.

4) The boundary points are classified into the cluster of the corresponding core points.

The specific number of clusters is automatically determined by the algorithm, which depends on the number of regions with different density characteristics in the data set.

C. Feature selection based on MIC theory

The number of features (variables), n, in a multivariate SCADA dataset usually ranges from dozens to hundreds. Nevertheless, it is not always the case that all features have a beneficial impact on WPF [38], and some even degrade the performance of prediction models. In addition, high memory consumption and model crashes may occur when all raw features enter into the Informer model. Therefore, to select the dominant features from SCADA data, it is desirable that the MIC theory is applied for feature selection and its basic computational process is outlined below.

Given variables $a=\{x_{1i},x_{2i},\cdots,x_{mi}\}$ and $b=\{y_1,y_2,\cdots,y_m\}$, representing i-th relevant feature series and wind power series, respectively. The mutual information (MI) between a and b is described below:

$$I(a,b)=\sum_{x\in a}\sum_{y\in b}p(x,y)\log_2\frac{p(x,y)}{p(x)p(y)} \tag{10}$$

where $p(x, y)$ represents the joint probability density function between a and b, and $p(x)$ and $p(y)$ are the respective marginal probability density functions.

Denote by $D(a, b)$ a dataset which has been divided into p and q segments along the X and Y axes. The maximum value of MI by dividing the $D(a, b)$ into $p \times q$ segments is given by:

$$I^*(D,p,q)=\max I(D|_g) \tag{11}$$

where $D|_g$ denotes the distribution of the dataset $D(a, b)$ on a grid g.

To facilitate the comparison and analysis of data in different units or scales, the data is normalized as follows:

$$M(D)_{p,q}=\frac{I^*(D,p,q)}{\log\min(p,q)} \tag{12}$$

The MIC value is calculated by dividing the grid with different $p \times q$ values.

$$MIC(D)=\max_{p,q<B(m)}\left[M(D)_{p,q}\right] \tag{13}$$

where $B(m)$ represents the upper limit of the grid division, which is generally given by $B(m) = m^{0.6}$ [39].

D. A novel periodic information enhanced embedding layer

To enhance the perception of the periodic variation characteristics of SCADA data and comprehensively analyze the relationship between various factors affecting wind power, a novel embedding layer construction method is proposed. The embedding layer is composed of scalar projection, local time stamp, and global time stamp.

Scalar projection: The TCN network, comprising multiple hidden layers, forms the scalar projection. This network is primarily composed of two key elements: dilated causal convolution and residual blocks. The dilated causal convolution exponentially expands the receptive field to accommodate a longer period of historical information while adhering to causal constraints. This convolution ensures that only neurons with outputs at time t, which depend on neurons at and before time t, are convolved. Different from traditional convolutional kernels, the kernel used for dilated causal convolution skips input samples at a fixed step size, which results in a wider coverage. Conceptually, an extended kernel can be conceptualized as a larger one, created by inserting zeros between adjacent kernel points. The dilated convolution operation F at time t of the one dimensional $X\in\mathbb{R}^n$ for a filter $f:\{0,1,...,k-1\}\to\mathbb{R}$ is defined as follows:

$$F(t)=(X*_d f)(t)=\sum_{i=0}^{k-1}f(i)\cdot x_{t-d\cdot i} \tag{14}$$

where the dilation factors are set to d = 1, 2, 4, while the filter size is designated as k = 2. In the constructed TCN scalar projection, three hidden layers are incorporated, with the neuron counts being 128, 256, and 512, respectively.

Furthermore, within each layer of the TCN, a residual module is incorporated. Weight normalization is applied to the dilated causal convolution, and regularization is facilitated through the introduction of spatial dropout. To guarantee the consistency of shape for the tensor utilized in elementwise addition, an additional 1 × 1 convolution is employed in the event that the input and output widths are incompatible.

Local time stamp: The local time stamp at time t in the sequence X_i is encoded as a fixed position in the sequence, as shown below:

$$P(p,j)=\begin{cases}\sin(p/(2L_x)^{j/d_{\text{modle}}}), j\ is\ even\\ \cos(p/(2L_x)^{j/d_{\text{modle}}}), j\ is\ odd\end{cases} \tag{15}$$

where p is the fixed position of the variable in the sequence, j = 1, 2, ⋯, d_{model}, L_x denotes the length of the sequence, d_{model} represents the dimension of hidden layer.

Global time stamp: Wind power data exhibits obvious time periodicity. In this study, periodic time elements such as minutes, hours, days, weeks and months are extracted as global time stamp. The global time stamp coding of moment t is defined as the date content in the X_t. Taking minutes as an example, the time coding of moment t is as follows:

$$M(t) = (\frac{m_t}{59} - 0.5) \tag{16}$$

where m_t denotes the minute date value at time t. Additionally, to minimize the contributions of varying dimensions, all dates are uniformly normalized to the interval [-0.5, 0.5]. To ensure the consistency of the global time stamp in dimensions, a linear layer is defined to linearly transform the content encoded in the global time stamp.

Finally, the scalar projection, local time stamp, and global time stamp are fused to form the input information for the informer wind power prediction model. Fig. 1 illustrates the embedding layer construction process.

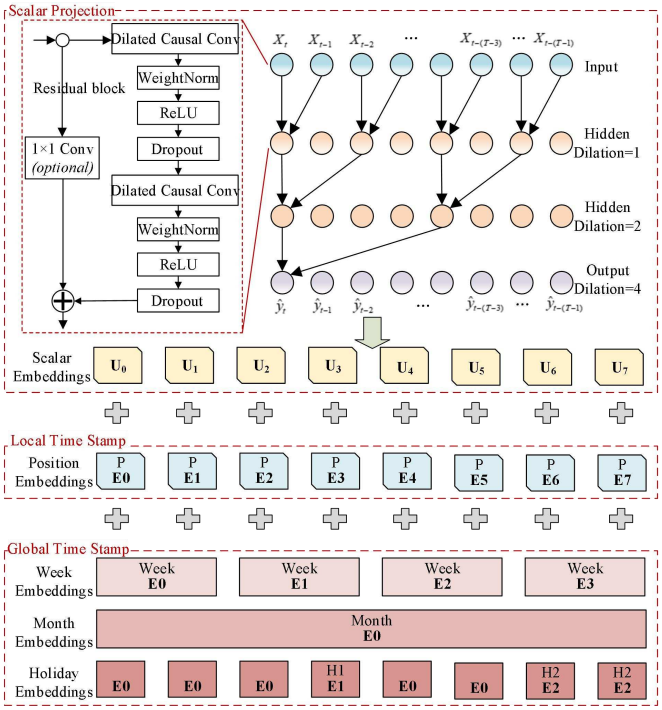

Fig. 1. Embedding layer construction for periodic information enhancement

E. A periodic-enhanced informer model for short-term WPF

A periodic-enhanced informer model for short-term WPF comprises three main components: an encoder and a decoder. These are principally constituted by probsparse self-attention, multi-head self-attention layer and multi-scale deep fusion module.

ProbSparse self-attention: The ProbSparse self-attention mechanism is schematically represented in Fig. 2. This mechanism reduces computational complexity to $O(L\log L)$ by restricting each key to attend only to the Top-u dominant queries, as determined by the following formula:

$$P_{robAttn}(Q,K,V) = Softmax(\frac{\bar{Q}K^T}{\sqrt{d_k}})V \tag{17}$$

where $Q \in \mathbb{R}^{L_Q \times d_k}$, $K \in \mathbb{R}^{L_k \times d_k}$, $V \in \mathbb{R}^{L_v \times d_k}$, $\bar{Q}$ represents a sparse matrix that exclusively incorporates the elements of Top-u Q satisfying the sparsity measurement criteria. K^T represents the transposition of K, d_k represents the input dimension. Set $u = c \times \ln L_Q$, where c is the sampling factor.

The sparsity criterion of the s-th row query vectors q_s is given by:

$$\bar{M}(q_s, K) = \max_j \left\{ \frac{q_s k_j^T}{\sqrt{d_k}} \right\} - \frac{1}{L_k} \sum_{j=1}^{L_k} \frac{q_s k_j^T}{\sqrt{d_k}} \tag{18}$$

where k_j represents j-th row K, $L_k = 1/q(k_j / q_s)$.

Multi-head self-attention layer: By utilizing a multi-head self-attention mechanism, information residing within diverse projection spaces is effectively acquired, ultimately enhancing the model feature extraction capability. Fig. 2 illustrates the structure of the multi-head self-attention layer. The model is fed the input X with linear projections through the matrices Q, K, and V. Then, these projections are forwarded to individual self-attention layer, resulting an output constructed from the H self-attention weighted values and the parameter matrix W^H, which is formulated as below:

$$M_{head}(Q,K,V) = Concat(Head_1, \cdots, Head_H)W^H \tag{19}$$

$$Head_h = Softmax(\frac{\bar{Q}W_{\bar{Q}}^h \left(KW_K^h\right)^T}{\sqrt{d_k}})VW_V^h \tag{20}$$

where $Head_h$ denotes the h-th self-attention weighted values. $W^H \in R^{d_k \times d_k}$, $W_K^h \in R^{N \times d_k}$, $W_{\bar{Q}}^h \in R^{N \times d_k}$, and $W_V^h \in R^{N \times d_k}$ are parameter matrices. $N = d_k/H$.

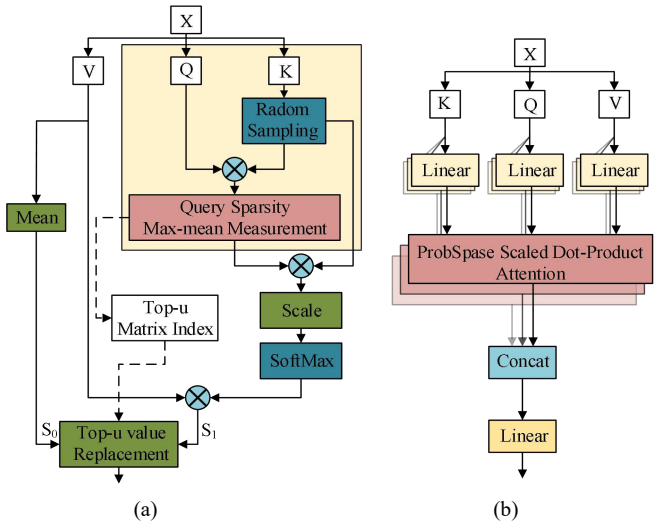

Fig. 2. Structures of two attention mechanisms. (a) ProbSparse self-attention. (b) Multi-head self-attention layer.

Multi-scale deep fusion module: To address the memory bottleneck associated when stacking layers, the distilling layer is employed. This operation prioritizes prominent high-quality features by utilizing CNN to generate a focused self-attention feature map for the succeeding layer. Following this, a down sampling process is performed using the maxpool operation to reduce the output length. However, relying solely on a single 1D-CNN cannot capture the multi-scale temporal structure

features existing in SCADA data. To overcome this limitation, the encoder and decoder add multiple maxpooling layers instead of simply deepening the network structure. This method expands the width of the network and avoids excessive resource consumption and overfitting problems. The structure is depicted in Fig. 3.

In order to achieve global features fusion, a 1D-CNN bottleneck layer with kernel 1 is introduced. By constructing a multi-scale maxpool auxiliary network, these pathways can share information with each other so that they can detect temporal characteristics at different scales. The minimum size of the maxpool layer is set to 3 to capture short term localized information, while the maximum size is set to 9 to capture long term information. Finally, the combination of local multi-scale and global features enhances the model's perception of dynamic sequence changes, thereby improving the accuracy and robustness of WPF. The improved operation from layer f to layer f+1 is given by:

$$P_{out1}^{t} = Conv1d\left(\left[X_{f}^{t}\right]_{AB}\right) \tag{21}$$

$$_{out2}^{t} = \sum_{k=1}^{4} Maxpool\left(\left(\left[X_{f}^{t}\right]_{AB}, \quad = 2k+1\right)\right) \tag{22}$$

$$X_{f+1}^{t} = Maxpool\left(ELU\left(P_{out1}^{t} + P_{out2}^{t}\right)\right) \tag{23}$$

where P_{out1}^{t} and P_{out2}^{t} denote the global features and local features at point t, respectively. K represents the size of the *Maxpool*. $\left[X_{f}^{t}\right]_{AB}$ represents the ProbSparse self-attention with multi-head.

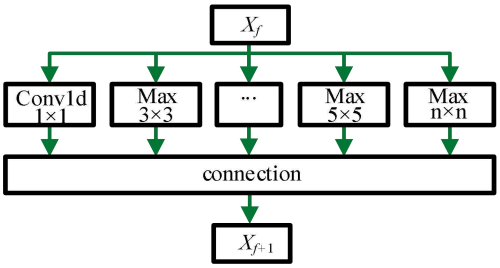

Fig. 3. A multi-scale deep fusion module

***Decoder*:** To tackle the challenge of predicting long outputs efficiently, a generative style decoder has been proposed. This decoder requires only one forward step, effectively mitigating the risk of cumulative error propagation during the inference stage. The output data X_{de}^{t} is as follows:

$$X_{de}^{t} = Concat\left(X_{token}^{t}, X_{0}^{t}\right) \in \mathbb{R}^{(L_{token}+L_{y})\times d_{model}} \tag{24}$$

where X_{token}^{t} is start token of the sequence, X_{0}^{t} represents the 0-valued placeholder. L_{token} and L_{y} are respectively the predicted and input sequence length.

III. Proposed Method

A. Network structure and training process

Taking the single stack in the proposed model encoder as an example, the internal structure of encoder can be shown in Fig. 4. The stack is a combination of embedding layer, multiple attention blocks and multi-scale deep fusion modules. After completing the data preprocessing, a two-dimensional multi-feature input set of dimension $L \times D$ will be generated, where L denotes length of the input sequence and D signifies the count of features. First, the input set is fed into the periodic information enhanced embedding layer to extract the hidden temporal characteristics. Secondly, a high-dimensional feature map is obtained through the multi-head probsparse self-attention. Finally, through the multi-scale deep fusion module, the global features are captured by 1D-CNN, and the local features are captured by multi-scale maxpool operation.

B. Framework of the suggested method

As depicted in Fig. 5, the suggested approach framework comprises three processes: data preprocessing, a periodic-enhanced informer model, and WPF.

In the process of data preprocessing, several essential steps are undertaken to generate suitable inputs for the model. Firstly, a method combining the quartile method and the DBSCAN algorithm is proposed to deal with outliers. Upon detecting the anomalous data points, the timestamps are immediately flagged, and all associated SCADA data for those timestamps are subsequently removed. Secondly, the MIC theory is utilized to construct a multi-feature input set that is strongly correlated with wind power, with aim of alleviating the curse of dimensionality and reducing the difficulty of the learning task. Subsequently, to maintain all the features within the same magnitude, normalization is applied, enhancing the model's convergence speed. Finally, the dataset is split into three parts: a train set, validation set and test set with a ratio of 7: 1: 2. The model training procedure is detailed in Section II. Upon completion of model training, the test set is inputted into the model for performance evaluation.

IV. Results and Discussion

A. Dataset preparation and experiment configuration

To evaluate the performance of the suggested approach, an actual SCADA dataset from wind farms in southern China was chosen as the benchmark. The wind farm comprises 25 wind turbines, with a total installed capacity of 50MW. The dataset used covers the entire year of 2020, with data sampled every 15 minutes. The relationship between output power P and the wind speed v of a wind turbine is effectively captured by the following power curve model:

$$P = \begin{cases} 0 & v < v_{\mathrm{i}}, v > v_{\mathrm{o}} \\ P(v) & v_{\mathrm{i}} \le v \le v_{\mathrm{r}} \\ P_{\mathrm{r}} & v_{\mathrm{r}} \le v \le v_{\mathrm{o}} \end{cases} \tag{25}$$

where v_{i} is the cut-in speed value of 3.5 m/s, v_{o} is the cut-out speed value of 25 m/s, v_{r} denotes the rated speed of the wind turbine at which it generates the rated power P_{r}. The values of v_{r} and P_{r} are 12 m/s and 2 MW, respectively.

The modelling experiments were conducted in Python 3.8 environment on system that was equipped with an Intel Core i7-13700KF CPU (64 GB RAM) and an NVIDIV RTX 4070S GPU.

B. Experimental settings and comparative methods

The experimental steps were divided into three parts: (i) A multi-feature input set construction based on MIC theory; (ii) Comparison of model performance at different time scales; (iii)

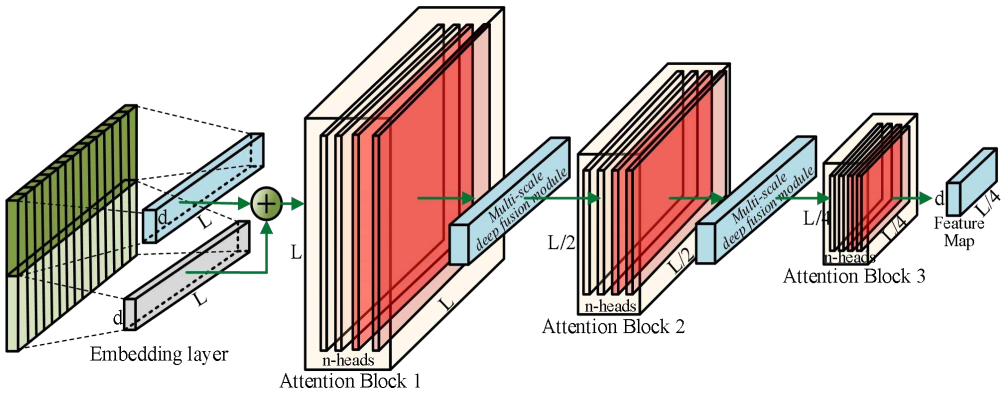

Fig. 4. The single stack in model's encoder.

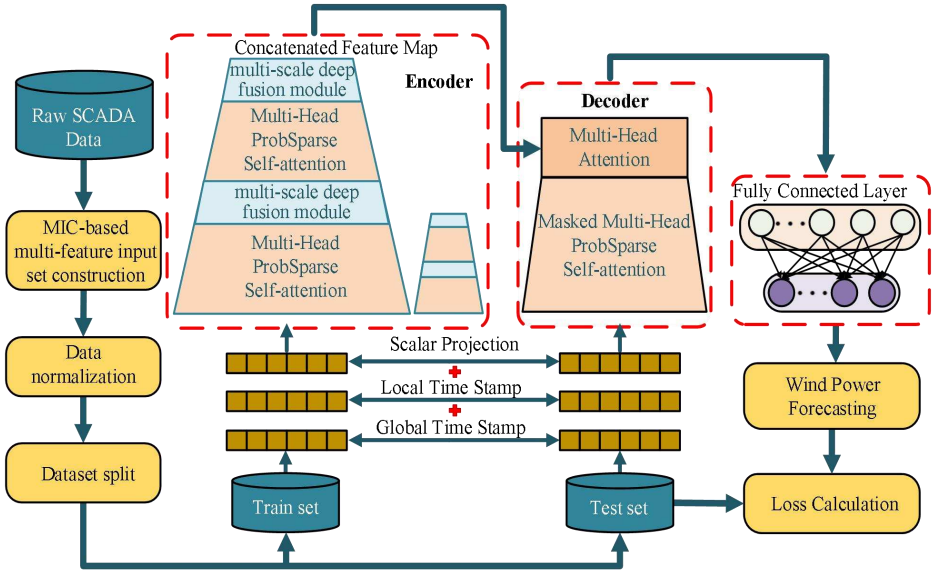

Fig. 5. Basic framework of the proposed model

Performance comparison of different models. To evaluate the performance of the proposed framework more accurately, commonly used models alongside variants of the Transformer were employed to predict up to 192 steps. The values of the hyper-parameters of the model are selected by a grid search [40]. The hyperparameters, along with their corresponding optimal values are briefly described in Table II. The learning_rate (LR) and batch_size have been identified as the most critical hyperparameters. The value of LR is typically defined within the range of 0.0001 to 0.1. To identify the optimal LR configuration, this study established a set of grid search values, specifically [0.1, 0.01, 0.001, 0.0001]. In regard to the batch size, it is essential to avoid extreme values, and it is recommended to select it based on the principle of 2^n, where n is chosen from the set [3, 4, 5, 6]. Additionally, the model parameter d_model is set to [128, 256, 512] to ensure a reasonable model structure. The remaining hyperparameters are configured according to the default values proposed by the associated original papers publicly available in the literature.

C. Evaluation metrics

This paper employed four commonly statistical metrics, namely mean absolute error (MAE), mean squared error (MSE), symmetric mean absolute percentage error (SMAPE) and root mean square error (RMSE) to assess the degree of fit between predicted and actual values. The formulas of the four metrics are as follows:

$$MAE = \frac{1}{n}\sum_{r=1}^{n}\left|P_r - \hat{P}_r\right| \tag{26}$$

$$MSE = \frac{1}{n}\sum_{r=1}^{n}(P_r - \hat{P}_r)^2 \tag{27}$$

$$SMAPE = \frac{1}{n}\sum_{r=1}^{n}\frac{|\hat{P}_r - P_r|}{(P_r + \hat{P}_r)/2} \tag{28}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{r=1}^{n}(P_r - \hat{P}_r)^2} \tag{29}$$

where P_r and $\hat{P}_r$ denote the actual and predicted value of the r-th wind power, n represents the count of predicted points.

D. A multi-feature input set construction based on MIC theory

To effectively filter out noise in SCADA data, a *v-p* curve-based method is adopted, where the quartile method is utilized to remove sparse outliers and the DBSCAN algorithm is employed to eliminate stacked outliers from the curve. Set ε to 0.015 and R to 4. From Fig. 6 (a) and (b), it can be observed that the combined method is remarkably effective in detecting outliers based on the quartiles and DBSCAN. Specifically, a total of 615 outliers have been identified. This includes 537 outliers identified based on the quartiles and 78 outliers identified using the DBSCAN algorithm. Further analysis of Fig. 6 (c) and (d) shows that the quartile method is primarily effective in eliminating sparse outliers, but falls short in identifying stacked outliers. In comparison, the DBSCAN algorithm is relatively more robust in identifying stacked outliers. However, outliers located around the *v-p* curve are not accurately detected. In SCADA data, upon identification of *v-p* anomalies, the corresponding timestamps are immediately flagged, and all associated SCADA data for those timestamps are subsequently removed.

TABLE II
MODEL HYPERPARAMETER NAMES AND THEIR LOCAL OPTIMAL VALUES

Model	Hyperparameters
Proposed method	d_model: 512, n_heads: 8, e_layers: 2, d_layers: 1, dropout: 0.05, learning_rete: 0.0001, batch_size: 32, activation: gelu
Informer	d_model: 512, n_heads: 8, e_layers: 2, d_layers: 1, dropout: 0.01, learning_rete: 0.0001, batch_size: 32, activation: gelu
Autoformer	d_model: 512, e_layers: 2, d_layers: 1, dropout: 0.01, learning_rete: 0.001, batch_size: 32, activation: gelu
Reformer	bucket_size: 32, n_hashes: 4, d_model: 512, batch_size: 32, e_layers: 2, d_layers: 1, learning_rete: 0.001, activation: gelu
Pyraformer	d_model: 512, e_layers: 2, d_layers: 1, dropout: 0.05, factor: 3, learning_rete: 0.001, batch_size: 16, activation: gelu
Crossformer	d_model: 256, e_layers: 2, d_layers: 2, dropout: 0.05, factor: 3, learning_rete: 0.01, batch_size: 16
ARIMA	p: 2, d: 1, q: 3
LSTM	units: 64, batch_size: 32, optimizer: adam, num_layers: 2, learning_rete: 0.0001
GRU	units: 128, batch_size: 32, optimizer: adam, num_layers: 3, learning_rete: 0.001

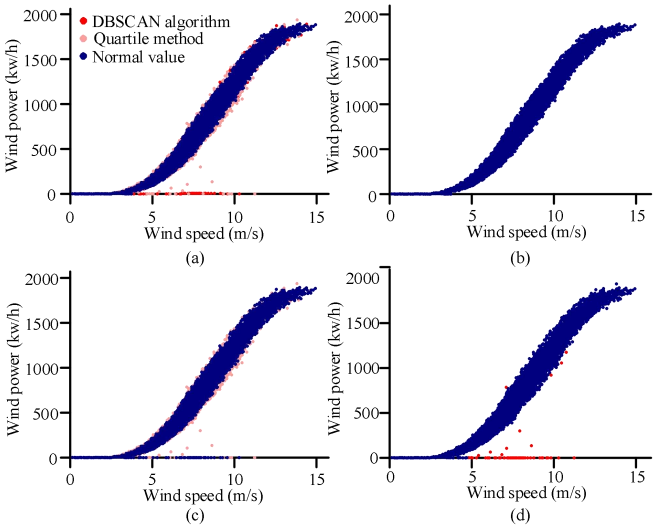

Fig. 6. Comparison of *v-p* scatter plots. (a) *v-p* scatter plot for outlier detection based on a combined algorithm. (b) *v-p* scatter plot of outliers removed. (c) *v-p* scatter plot for outlier detection based on the quartile method. (d) *v-p* scatter plot for outlier detection based on the DBSCAN method.

The SCADA system of the selected wind farms collects a substantial amount of operational data, comprising twenty relevant features and one target feature after preliminary filtering. Taking the current wind power as a label, the MIC values were calculated, and features exhibiting high correlation were used for the multi-feature input set. Table II presents the MIC values of the relevant features alongside wind power.

As indicated in Table III, the MIC values of the following variables are all over 0.8: *rotor speed*, *speed detection value of overspeed sensor*, *generator operating frequency*, *generator current*, *generator torque*, *wind speed*, *current of pitch motor* and *inverter INU temperature* exhibit high correlation with wind power. The *estimated power of pitch motor* and *angle of blade 1* have a certain relation, with respective MIC values of 0.7221 and 0.6627. Conversely, the *Y-direction vibration value* and *inverter inlet pressure* are relatively small, with MIC values below 0.3.

TABLE III
THE MIC VALUES OF RELEVANT FEATURES

Monitoring parameters	MIC	Monitoring parameters	MIC
Rotor speed	0.9397	Angle of blade 2	0.6528
Speed detection value of overspeed sensor	0.9276	Angle of blade 3	0.6511
Generator operating frequency	0.9228	Generator stator temperature	0.5875
Generator current	0.9190	Main bearing temperature 1	0.5072
Generator torque	0.9131	Main bearing temperature 2	0.4372
Wind speed	0.9120	Hydraulic braking pressure	0.3802
Current of pitch motor	0.8791	Y-direction vibration value	0.2205
Inverter INU temperature	0.8215	X-direction vibration value	0.1639
Estimated power of pitch motor	0.7221	Inverter inlet pressure	0.1557
Angle of blade 1	0.6627	Inverter outlet pressure	0.1334

Fig. 7 describes the changes of wind power and relevant features over time using a line graph. As shown in the figure, *rotor speed* and *speed detection value of overspeed sensor* exhibit the highest correlation with wind power, and *current of pitch motor* and *inverter INU temperature* also exhibit positive correlation. These features are highly related with the target value, and therefore provide useful information for building WPF model. Based on the aforementioned analysis, features with wind power MIC values over 0.8 were selected to construct the multi-feature input set.

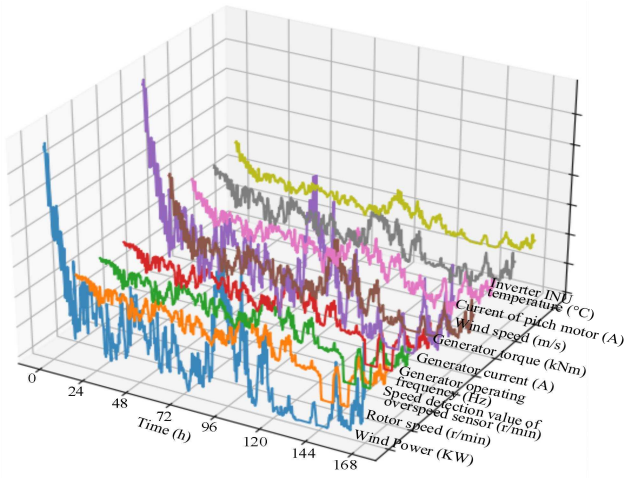

Fig. 7. The hourly variations of wind power and the relevant features over a continuous week in January 2020

E. Comparison of model performance at different time scales

To verify the effectiveness of the multi-scale deep fusion module in wind power prediction, four models with different number of maxpool scales were constructed, and multi-step prediction experiments with steps of 12, 24, 48, 96 and 196were carried out. The results are displayed in Table III, where the performance of the model was evaluated using four metrics (MSE, MAE, RMSE, SMAPE). The most outstanding outcomes are emphasized in bold font.

TABLE IV
COMPARISON OF METRICS AT DIFFERENT TIME SCALES

Step	Metric	One scale	Two scales	Three scales	Four scales	Informer	ablation
12	MSE	0.0246	0.0206	0.0180	0.0187	0.0210	**0.0171**
	MAE	0.1198	0.1107	0.1027	0.1047	0.1089	**0.1004**
	RMSE	0.1570	0.1435	0.1341	0.1367	0.1450	**0.1309**
	SMAPE	0.3474	0.3068	**0.2957**	0.3083	0.3179	0.3176
24	MSE	0.0291	0.0257	**0.0220**	0.0253	0.0271	0.0257
	MAE	0.1304	0.1223	**0.1138**	0.1203	0.1264	0.1187
	RMSE	0.1706	0.1603	**0.1484**	0.1590	0.1645	0.1602
	SMAPE	0.3504	0.3319	**0.3182**	0.3464	0.3513	0.3380
48	MSE	0.0386	0.0409	**0.0339**	0.0353	0.0414	0.0367
	MAE	0.1528	0.1546	**0.1428**	0.1449	0.1580	0.1476
	RMSE	0.1965	0.2022	**0.1841**	0.1878	0.2034	0.1916
	SMAPE	0.4018	0.4177	**0.3868**	0.4087	0.4393	0.4105
96	MSE	0.0585	**0.0497**	0.0500	0.0542	0.0572	0.0596
	MAE	0.1915	**0.1734**	0.1746	0.1849	0.1922	0.1899
	RMSE	0.2418	**0.2228**	0.2237	0.2328	0.2391	0.2442
	SMAPE	0.5014	**0.4553**	0.4578	0.4846	0.4998	0.5304
192	MSE	0.0867	0.0849	**0.0682**	0.0732	0.0793	0.0817
	MAE	0.2416	0.2276	**0.2109**	0.2173	0.2311	0.2234
	RMSE	0.2944	0.2913	**0.2612**	0.2705	0.2815	0.2858
	SMAPE	0.6164	0.5983	0.5446	0.5723	**0.5430**	0.5806

TABLE V
COMPARISON OF METRICS IN DIFFERENT MODELS

Step	Metric	Proposed method	Informer	Autoformer	Reformer	Pyraformer	Crossformer	ARIMA	LSTM	GRU
12	MSE	**0.0180**	0.0210	0.0284	0.0280	0.0252	0.0196	0.0249	0.0484	0.0381
	MAE	**0.1027**	0.1089	0.1308	0.1217	0.1193	0.1068	0.1127	0.1779	0.1482
	RMSE	**0.1341**	0.1450	0.1684	0.1673	0.1587	0.1401	0.1579	0.2201	0.1952
	SMAPE	**0.2957**	0.3179	0.4049	0.4573	0.3435	0.3142	0.3049	0.5273	0.3878
24	MSE	**0.0220**	0.0271	0.0378	0.0471	0.0351	0.0268	0.0467	0.0533	0.0494
	MAE	**0.1138**	0.1264	0.1531	0.1649	0.1414	0.1256	0.1846	0.2005	0.1928
	RMSE	**0.1484**	0.1645	0.1945	0.2171	0.1874	0.1637	0.2161	0.2309	0.2223
	SMAPE	**0.3182**	0.3613	0.4215	0.5682	0.4023	0.3534	0.4944	0.4693	0.4772
48	MSE	**0.0339**	0.0414	0.0563	0.0631	0.0488	0.0437	0.0593	0.0727	0.0800
	MAE	**0.1428**	0.1580	0.1902	0.2070	0.1705	0.1606	0.2108	0.2317	0.2423
	RMSE	**0.1841**	0.2034	0.2372	0.2512	0.2210	0.2090	0.2435	0.2696	0.2828
	SMAPE	**0.3868**	0.4393	0.5080	0.6249	0.4637	0.4342	0.4629	0.4604	0.4621
96	MSE	**0.0500**	0.0572	0.0666	0.0780	0.0623	0.0780	0.0914	0.0878	0.0927
	MAE	**0.1746**	0.1922	0.2105	0.2225	0.2003	0.2245	0.2586	0.2535	0.2605
	RMSE	**0.2237**	0.2391	0.2580	0.2792	0.2496	0.2794	0.3023	0.2963	0.3044
	SMAPE	**0.4578**	0.4998	0.5333	0.6586	0.5198	0.5833	0.4677	0.4658	0.4685
192	MSE	**0.0682**	0.0793	0.0821	0.0854	0.0847	0.0848	0.1145	0.1430	0.2169
	MAE	**0.2109**	0.2311	0.2380	0.2385	0.2277	0.2380	0.2841	0.3300	0.4281
	RMSE	**0.2612**	0.2815	0.2866	0.2922	0.2792	0.2912	0.3384	0.3782	0.4657
	SMAPE	0.5446	**0.5430**	0.5549	0.7070	0.6442	0.6096	0.4921	0.5295	0.6240

As evidenced from Table IV, the suggested approach has better performance than the original Informer. The prediction task performs well for the cases with three and more maxpool scales. When the number of scales is 3, compared to the informer model, the MAE value decreases by 9.97% (from 0.1264 to 0.1138) at 24-step and 8.74% (from 0.2311 to 0.2109) at 192-step; the RMSE value decreases by 9.79% (from 0.1645 to 0.1484) at 24-step and 7.21% (from 0.2815 to 0.2612) at 192-step. At 96-step, the better prediction effect is achieved when the number of scales is 2. However, compared with the scale number of 3, the MAE value decreases by 0.69% (from 0.1746 to 0.1734); the RMSE value decreases by 0.40% (from 0.2237 to 0.2228). In summary, when the number of scales is small, the model cannot fully exploit the intrinsic correlation of the SCADA system data, whereas a larger number of scales requires more computational resources. The proposed model has better overall performance at scale 3.

The effectiveness of using the novel periodic information enhanced embedding layer was demonstrated through ablation experiment, specifically by removing the multi-scale deep fusion module from the network model structure commonly used in many existing methods. It turns out that the ablation experiment results are better than the informer model. When the number of scales is 1 or 2, the proposed model does not perform as well as the ablative experimental model, but the advantage comes to the fore by the number of scales of 3 or more. Compared with the ablation experiment, when the number of scales is 3, MAE is reduced by 8.07% (0.1899 from

to 0.1746) and RMSE is reduced by 8.39% (0.2442 from to 0.2237) at 96-step, MAE is reduced by 5.60% (0.2234 from to 0.2109) and RMSE is increased by 8.61% (0.2858 from to 0.2612) at 192-step.

F. Performance comparison of different models

In this paper, three commonly used time-series forecasting methods, ARIMA, LSTM and GRU, were applied to the same data and the performances are compared. In order to further explore the performance of transformer for wind power time-

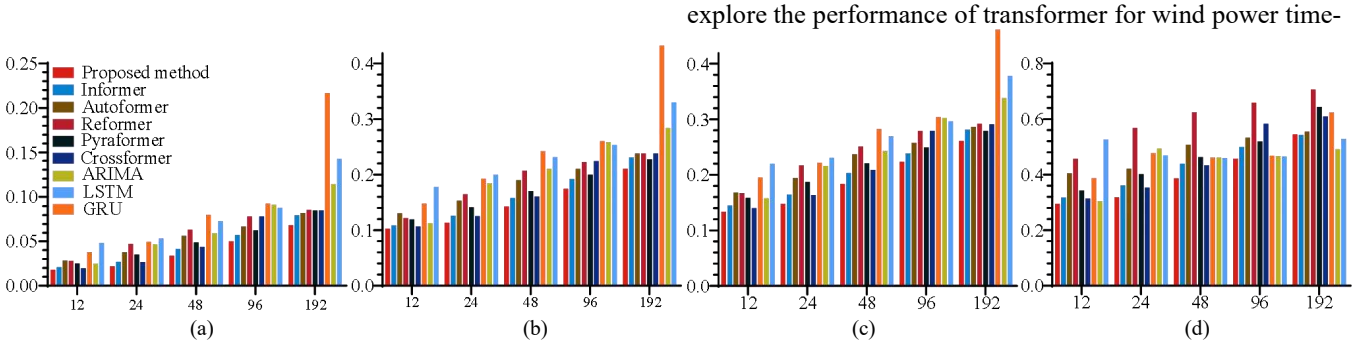

Fig. 8. Prediction results of all the methods. (a) MSE values. (b) MAE values. (c) RMSE values. (d) SMAPE values.

TABLE VI
COMPARISON OF MSE AND MAE METRICS IN DIFFERENT SEASONS

Season		Proposed method		Informer		Autoformer		Reformer		Pyraformer		Crossformer	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Spring	12	**0.0122**	**0.0683**	0.0129	0.0747	0.0223	0.1099	0.0176	0.0909	0.0146	0.0762	0.0144	0.0792
	24	0.0250	0.1061	**0.0194**	0.1077	0.0349	0.1440	0.0240	0.1160	0.0268	0.1077	0.0208	**0.1020**
	48	0.0258	**0.1121**	**0.0230**	0.1127	0.0374	0.1383	0.0293	0.1253	0.0325	0.1271	0.0371	0.1537
	96	**0.0337**	**0.1394**	0.0345	0.1503	0.0456	0.1555	0.0346	0.1427	0.0411	0.1493	0.0384	0.1596
	192	0.0380	0.1568	**0.0360**	0.1586	0.0435	0.1639	0.0373	**0.1454**	0.0461	0.1534	0.0383	0.1549
Summer	12	**0.0164**	**0.0958**	0.0171	0.0990	0.0217	0.1158	0.0177	0.1013	0.0199	0.1095	0.0175	0.1013
	24	**0.0180**	**0.1009**	0.0236	0.1157	0.0263	0.1251	0.0246	0.1190	0.0231	0.1178	0.0226	0.1145
	48	**0.0241**	**0.1180**	0.0263	0.1272	0.0281	0.1356	0.0276	0.1306	0.0274	0.1299	0.0287	0.1296
	96	0.0359	0.1475	0.0400	0.1543	0.0313	0.1385	0.0374	0.1473	0.0319	0.1416	**0.0309**	**0.1376**
	192	0.0420	0.1602	0.0449	0.1666	0.0431	0.1614	0.0384	0.1532	**0.0304**	**0.1365**	0.0409	0.1589
Autumn	12	**0.0049**	**0.0467**	0.0054	0.0474	0.0086	0.0673	0.0051	0.0470	0.0057	0.0489	0.0155	0.0982
	24	**0.0063**	**0.0502**	0.0087	0.0653	0.0118	0.0776	0.0072	0.0540	0.0080	0.0577	0.0167	0.1074
	48	**0.0070**	**0.0545**	0.0101	0.0637	0.0159	0.0906	0.0110	0.0665	0.0114	0.0719	0.0177	0.1135
	96	0.0171	0.0992	0.0172	0.0979	0.0182	0.0944	0.0150	0.0867	**0.0125**	**0.0748**	0.0218	0.1232
	192	0.0211	0.1110	0.0193	0.1011	0.0169	0.1095	0.0194	0.1043	**0.0157**	**0.0913**	0.0252	0.1345
Winter	12	**0.0179**	**0.0983**	0.0197	0.1035	0.0301	0.1314	0.0263	0.1280	0.0283	0.1230	0.0250	0.1189
	24	**0.0265**	**0.1197**	0.0331	0.1367	0.0514	0.1763	0.0517	0.1807	0.0414	0.1519	0.0391	0.1461
	48	**0.0369**	**0.1431**	0.0603	0.1990	0.0711	0.2193	0.0592	0.1904	0.0615	0.1932	0.0522	0.1771
	96	**0.0541**	**0.1873**	0.0655	0.2186	0.0835	0.2387	0.0799	0.2269	0.0788	0.2286	0.0619	0.2069
	192	**0.0611**	**0.2049**	0.0733	0.2286	0.0793	0.2321	0.1169	0.2765	0.0908	0.2370	0.0637	0.2064

TABLE VII
COMPARISON OF RMSE AND SMAPE METRICS IN DIFFERENT SEASONS

Season		Proposed method		Informer		Autoformer		Reformer		Pyraformer		Crossformer	
		RMSE	SMAPE	RMSE	SMAPE	RMSE	SMAPE	RMSE	SMAPE	RMSE	SMAPE	RMSE	SMAPE
Spring	12	**0.1104**	**0.6879**	0.1135	0.7790	0.1493	0.9934	0.1326	0.8361	0.1207	0.8495	0.1198	0.8800
	24	0.1582	**0.8658**	**0.1394**	0.9020	0.1869	1.0367	0.1548	0.9252	0.1638	0.9780	0.1441	0.9095
	48	0.1608	**0.8968**	**0.1518**	0.9066	0.1934	1.0253	0.1710	0.9558	0.1801	0.9984	0.1927	1.0013
	96	**0.1837**	1.0384	0.1857	0.9969	0.2135	1.1052	0.1859	**0.9916**	0.2027	1.0852	0.1960	1.0016
	192	0.1949	1.0403	**0.1898**	**0.9874**	0.2085	1.0685	0.1932	1.0216	0.2147	1.1559	0.1956	1.0161
Summer	12	**0.1281**	**0.3633**	0.1308	0.3751	0.1472	0.4312	0.1329	0.3804	0.1412	0.4051	0.1325	0.3853
	24	**0.1341**	**0.3823**	0.1536	0.4312	0.1622	0.4805	0.1569	0.4384	0.1519	0.4314	0.1503	0.4247
	48	**0.1553**	**0.4457**	0.1623	0.4639	0.1676	0.4849	0.1660	0.4583	0.1655	0.4706	0.1693	0.4850
	96	0.1895	0.5666	0.1999	0.5941	0.1769	**0.4963**	0.1933	0.4992	0.1786	0.5066	**0.1758**	0.5063
	192	0.2049	0.6310	0.2119	0.6493	0.2077	0.6119	0.1960	0.5485	**0.1743**	**0.4816**	0.2022	0.5868
Autumn	12	**0.0703**	**0.5260**	0.0734	0.5491	0.0930	0.7917	0.0717	0.5459	0.0757	0.5472	0.1247	0.8378
	24	**0.0794**	**0.5667**	0.0931	0.6343	0.1085	0.8721	0.0847	0.5973	0.0892	0.6065	0.1294	0.8641
	48	**0.0839**	**0.6096**	0.1007	0.6627	0.1259	0.9058	0.1051	0.7254	0.1069	0.7268	0.1331	0.8856
	96	0.1308	0.7961	0.1308	0.9388	0.1349	0.9476	0.1226	0.8613	**0.1118**	**0.7665**	0.1476	0.9233
	192	0.1452	0.8900	0.1390	0.9211	0.1298	0.8838	0.1393	0.9486	**0.1254**	**0.8569**	0.1589	0.9405
Winter	12	**0.1339**	0.3760	0.1405	0.3971	0.1734	0.4913	0.1622	**0.3660**	0.1684	0.4412	0.1580	0.4308
	24	**0.1627**	**0.4348**	0.1818	0.4852	0.2267	0.5781	0.2276	0.4734	0.2034	0.5484	0.1978	0.5433
	48	**0.1920**	**0.4923**	0.2456	0.6100	0.2666	0.6409	0.2433	0.6300	0.2480	0.6270	0.2285	0.5763
	96	**0.2326**	**0.5852**	0.2560	0.6416	0.2890	0.6770	0.2826	0.5881	0.2808	0.7039	0.2487	0.6138
	192	**0.2472**	**0.6463**	0.2707	0.6523	0.2816	0.7006	0.3419	0.7385	0.3014	0.7036	0.2523	0.6257

series forecasting, several effective variants were introduced in the experiment: Reformer, Autoformer, Pyraformer, and the latest variant Crossformer. Table V demonstrates results of these models, where the smallest value of error is marked in bold. The error comparison histograms are shown in Fig. 8.

As illustrated in Table V and Fig. 8, transformer-based time-series forecasting demonstrates superior performance compared to traditional methods, including LSTM and GRU, for short-term prediction problems. The proposed model has achieved relatively good results compared with the variants of Transformer. To further evaluate the robustness of the proposed model, this study performed comprehensive analysis of WPF across various seasonal conditions. The results are shown in VI and VII. Taking the MSE and MAE as an example, the proposed model shows overall superiority. For winter wind forecast, the proposed method outperforms all the compared methods within the forecast range spanning from 12 steps to 192 steps. The spring prediction results indicate that the achieved MSE and MAE values are distributed irregularly under the specified step size. For summer and autumn wind forecasts, the Pyraformer model performs better than other variants when the forecast range exceeds 96 steps. In summer, compared with the proposed model, the MAE value of Pyraformer decreased by 14.79% (from 0.1602 to 0.1365) at 192 steps and the MSE value decreased by 27.62% (from 0.0420 to 0.0304).

V. Conclusions

With the wide application of SCADA systems in wind farms, the operating data and environmental data of wind turbines show an explosive growth trend. These rich datasets are increasingly valuable for WPF analysis. Consequently, this paper proposes a period-enhanced informer model, which aims to effectively utilize SCADA data for short-term WPF. The model is mainly composed of the following parts: (1) Based on the MIC theory, a multi-feature input set is constructed to comprehensively consider the factors affecting wind power while reducing the memory consumption burden. (2) A novel periodic information enhanced embedding layer is designed to extract the periodic variation characteristics hidden in SCADA data. (3) To extract a broader range of time-series features, a multi-scale deep fusion module is proposed and incorporated into the Informer model. Although the proposed model can help improve prediction accuracy in comparison with other models, future research can be done to further improve its performance in the following aspects:

1) Data denoising. This study only constructed a multi-feature input set but did not perform data enhancement. In the future, research work will be undertaken to develop denoising methods to improve the quality of SCADA data.
2) Lightweight mode. Although the periodic information enhanced embedding layer and the multi-scale fusion module have improved Informer's predictive capabilities, these components have also led to an increase in runtime. In the future work, the model will be optimized from the perspective of model lightweight.
3) Probability interval prediction. The model presented in this paper is exclusively designed for point prediction, while the consideration of probability interval prediction is reserved for future exploration.

References

[1] C. Zou, Q. Zhao, G. Zhang, and B. Xiong, "Energy revolution: From a fossil energy era to a new energy era," *Natural Gas Industry B,* vol. 3, no. 1, pp. 1-11, 2016.

[2] M. J. Sanjari, H. B. Gooi, and N.-K. C. Nair, "Power generation forecast of hybrid PV–wind system," *IEEE Transactions on Sustainable Energy,* vol. 11, no. 2, pp. 703-712, 2019.

[3] M. Khodayar, and J. Wang, "Spatio-temporal graph deep neural network for short-term wind speed forecasting," *IEEE Transactions on Sustainable Energy,* vol. 10, no. 2, pp. 670-681, 2018.

[4] J. Ding, K. Xie, B. Hu, C. Shao, T. Niu, C. Li, and C. Pan, "Mixed aleatory-epistemic uncertainty modeling of wind power forecast errors in operation reliability evaluation of power systems," *Journal of Modern Power Systems and Clean Energy,* vol. 10, no. 5, pp. 1174-1183, 2022.

[5] H. Zhang, J. Yan, Y. Liu, Y. Gao, S. Han, and L. Li, "Multi-source and temporal attention network for probabilistic wind power prediction," *IEEE Transactions on Sustainable Energy,* vol. 12, no. 4, pp. 2205-2218, 2021.

[6] Y. Chang, H. Yang, Y. Chen, M. Zhou, H. Yang, Y. Wang, and Y. Zhang, "A Hybrid Model for Long-Term Wind Power Forecasting Utilizing NWP Subsequence Correction and Multi-Scale Deep Learning Regression Methods," *IEEE Transactions on Sustainable Energy*, 2023.

[7] Z. Sun, and M. Zhao, "Short-term wind power forecasting based on VMD decomposition, ConvLSTM networks and error analysis," *IEEE Access,* vol. 8, pp. 134422-134434, 2020.

[8] Y. Dong, S. Ma, H. Zhang, and G. Yang, "Wind power prediction based on multi-class autoregressive moving average model with logistic function," *Journal of Modern Power Systems and Clean Energy,* vol. 10, no. 5, pp. 1184-1193, 2022.

[9] A. Meng, S. Chen, Z. Ou, W. Ding, H. Zhou, J. Fan, and H. Yin, "A hybrid deep learning architecture for wind power prediction based on bi-attention mechanism and crisscross optimization," *Energy,* vol. 238, pp. 121795, 2022.

[10] W. Zhang, Z. Lin, and X. Liu, "Short-term offshore wind power forecasting-A hybrid model based on Discrete Wavelet Transform (DWT), Seasonal Autoregressive Integrated Moving Average (SARIMA), and deep-learning-based Long Short-Term Memory (LSTM)," *Renewable Energy,* vol. 185, pp. 611-628, 2022.

[11] M.-S. Ko, K. Lee, J.-K. Kim, C. W. Hong, Z. Y. Dong, and K. Hur, "Deep concatenated residual network with bidirectional LSTM for one-hour-ahead wind power forecasting," *IEEE Transactions on Sustainable Energy,* vol. 12, no. 2, pp. 1321-1335, 2020.

[12] Q. Hu, S. Zhang, M. Yu, and Z. Xie, "Short-term wind speed or power forecasting with heteroscedastic support vector regression," *IEEE Transactions on Sustainable Energy,* vol. 7, no. 1, pp. 241-249, 2015.

[13] Y. Chen, H. Zhao, R. Zhou, P. Xu, K. Zhang, Y. Dai, H. Zhang, J. Zhang, and T. Gao, "CNN-BiLSTM short-term wind power forecasting method based on feature selection," *IEEE Journal of Radio Frequency Identification,* vol. 6, pp. 922-927, 2022.

[14] G. Memarzadeh, and F. Keynia, "A new short-term wind speed forecasting method based on fine-tuned LSTM neural network and optimal input sets," *Energy Conversion and Management,* vol. 213, pp. 112824, 2020.

[15] R. Zhu, W. Liao, and Y. Wang, "Short-term prediction for wind power based on temporal convolutional network," *Energy Reports,* vol. 6, pp. 424-429, 2020.

[16] J. Liang, and W. Tang, "Ultra-short-term spatiotemporal forecasting of renewable resources: An attention temporal convolutional network-based approach," *IEEE Transactions on Smart Grid,* vol. 13, no. 5, pp. 3798-3812, 2022.

[17] Y. Xiao, S. Wu, C. He, Y. Hu, and M. Yi, "An effective hybrid wind power forecasting model based on" decomposition-reconstruction-ensemble" strategy and wind resource matching," *Sustainable Energy, Grids and Networks,* vol. 38, pp. 101293, 2024.

[18] X. Deng, H. Shao, C. Hu, D. Jiang, and Y. Jiang, "Wind power forecasting methods based on deep learning: A survey," *Computer Modeling in Engineering and Sciences,* vol. 122, no. 1, pp. 273, 2020.

[19] Z. Lin, and X. Liu, "Wind power forecasting of an offshore wind turbine based on high-frequency SCADA data and deep learning neural network," *Energy,* vol. 201, pp. 117693, 2020.

[20] R. Morrison, X. Liu, and Z. Lin, "Anomaly detection in wind turbine SCADA data for power curve cleaning," *Renewable Energy,* vol. 184, pp. 473-486, 2022.

[21] X. Liu, L. Yang, and Z. Zhang, "Short-term multi-step ahead wind power predictions based on a novel deep convolutional recurrent network method," *IEEE Transactions on Sustainable Energy,* vol. 12, no. 3, pp. 1820-1833, 2021.

[22] Z.-H. Liu, C.-T. Wang, H.-L. Wei, B. Zeng, M. Li, and X.-P. Song, "A wavelet-LSTM model for short-term wind power forecasting using wind farm SCADA data," *Expert Systems with Applications,* vol. 247, pp. 123237, 2024.

[23] Z. Lin, X. Liu, and M. Collu, "Wind power prediction based on high-frequency SCADA data along with isolation forest and deep learning neural networks," *International Journal of Electrical Power & Energy Systems,* vol. 118, pp. 105835, 2020.

[24] L. Wang, Y. He, L. Li, X. Liu, and Y. Zhao, "A novel approach to ultra-short-term multi-step wind power predictions based on encoder–decoder architecture in natural language processing," *Journal of Cleaner Production,* vol. 354, pp. 131723, 2022.

[25] Y. Cao, G. Liu, D. Luo, D. P. Bavirisetti, and G. Xiao, "Multi-timescale photovoltaic power forecasting using an improved Stacking ensemble algorithm based LSTM-Informer model," *Energy,* vol. 283, pp. 128669, 2023.

[26] S. Zheng, and J. Liu, "Automatic Multi-steps Prediction Modelling for Wind Power Forecasting." pp. 133-139.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems,* vol. 30, 2017.

[28] C. Bi, P. Ren, T. Yin, Y. Zhang, B. Li, and Z. Xiang, "An informer architecture-based ionospheric foF2 model in the middle latitude region," *IEEE Geoscience and Remote Sensing Letters,* vol. 19, pp. 1-5, 2022.

[29] R. Child, S. Gray, A. Radford, and I. Sutskever, "Generating long sequences with sparse transformers," *arXiv preprint arXiv:1904.10509*, 2019.

[30] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in neural information processing systems,* vol. 34, pp. 22419-22430, 2021.

[31] S. Liu, H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu, and S. Dustdar, "Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting." in *International conference on learning representations,* 2020.

[32] J. Qiu, H. Ma, O. Levy, W.-t. Yih, S. Wang, and J. Tang, "Blockwise Self-Attention for Long Document Understanding." *arXiv preprint arXiv:1911.02972*, 2019.

[33] N. Kitaev, L. Kaiser, and A. Levskaya, "Reformer: The Efficient Transformer." *arXiv preprint arXiv:2001.04451*, 2020.

[34] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM computing surveys (CSUR),* vol. 54, no. 10s, pp. 1-41, 2022.

[35] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting." in *Proceedings of the AAAI conference on artificial intelligence*, 2021.

[36] X. Liao, Z. Liu, X. Zheng, Z. Ping, and X. He, "Wind power prediction based on periodic characteristic decomposition and multi-layer attention network," *Neurocomputing,* vol. 534, pp. 119-132, 2023.

[37] S. Swapna, P. Niranjan, B. Srinivas, and R. Swapna, "Data cleaning for data quality." *IEEE International Conference on Computing for Sustainable Global Development*, pp. 344-348, 2016.

[38] Y. He, J. Liu, S. Wu, and X. Wang, "Condition monitoring and fault detection of wind turbine driveline with the implementation of deep residual long short-term memory network," *IEEE Sensors Journal*, 2023.

[39] C. Liu, S. Wang, H. Yuan, and X. Liu, "Detecting unbiased associations in large data sets," *Big Data,* vol. 10, no. 4, pp. 337-355, 2022.

[40] F. Chen, J. Yan, L. B. Tjernberg, D. Song, Y. Yan, and Y. Liu, "Medium-Term Wind Power Forecasting based on Dynamic Self-Attention Mechanism." *IEEE Belgrade PowerTech*, pp. 1-5, 2023.

Zhao-Hua Liu (M'16，SM'2022) He received M.Sc. degree in computer science and engineering, and the PhD. degree in automatic control and electrical engineering from the Hunan University, China, in 2010 and 2012, respectively. He worked as a visiting researcher in the Department of Automatic Control and Systems Engineering at the University of Sheffield, United Kingdom, from 2015 to 2016.

He is currently a Professor with the School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan, China. His current research interests include computational intelligence and learning algorithms design, parameter estimation and control of permanent-magnet synchronous machine drives,wind power forecasting and wind turbine intelligent control, and condition monitoring and fault diagnosis for electrical equipment.

Dr. Liu has published a monograph in the field of Biological immune system inspired hybrid intelligent algorithm and its applications, and published more than 60 research papers in refereed journals and conferences, including IEEE TRANSACTIONS/JOURNAL/MAGAZINE. He is a regular reviewer for several international journals and conferences.

Long-Wei Li received B. Eng. degree in Automation from the Henan Polytechnic University, Jiaozuo, China, in 2022. He is currently pursuing the M.S. degree in Control Science and Engineering, at Hunan University of Science and Technology, Xiangtan, China.

His current research interests include deep learning algorithm design and wind power forecasting.

Hua-Liang Wei received the Ph.D. degrees in automatic control from The University of Sheffield, Sheffield, U.K., in 2004.

He is currently a senior lecturer with the Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, UK. His research focuses on evolutionary algorithms, identification and modelling for complex nonlinear systems, applications and developments of signal processing, system identification and data modelling to control engineering.

Ming Li received the B.S. degree in electrical engineering and automation from China Three Gorges University, Yichang, China, in 2013, and the M.E. degree in power engineering from the Hunan Institute of Engineering, Xiangtan, China, in 2016, and the Ph.D. degree in in computer science and technology from the Hunan University, China, in 2023.

She is currently a Lecturer with the School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan, China. Her current research interests include parameter/state estimation of motor drive system, and wind power forecasting and intelligent control.

Ming-Yang Lv received the Ph.D. degree in control theory and engineering with Hunan University, Changsha, China, in 2020.

He is currently a Lecturer with the School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan, China. His research interests include chaos theory and application, and modeling for complex process industries based on machine learning and deep learning.

Ying-Jie Zhang received the Ph.D. degree in control theory and control engineering from Hunan University (HNU), Changsha, China, in 2005. From 2010 to 2011, he was a Visiting Scholar with the University of Oslo, Oslo, Norway. In 2018, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, Technische Universität Dresden, Dresden, Germany.

He is currently a Professor and the Yuelu Scholar with the College of Computer Science and Electronic Engineering, HNU, where he is also the Director of the Institute of Industry Energy-Saving Control and Evaluation. His research interests include parameter/state estimation, modeling, intelligent control, energy optimization, and intelligent control with applications to hybrid electric and autonomous vehicles, and fault detection and diagnosis for rotating machines.