An End-to-End Ensemble Learning Approach for Enhancing Wind Power Forecasting

Yun Wang, Houhua Xu, Yaohui Huang, Fan Zhang, Hongbo Kou, Runmin Zou, Qinghua Hu, *Senior Member, IEEE*, Dipti Srinivasan, *Fellow, IEEE*

***Abstract*—Accurate wind power forecasts are crucial for grid stability, thereby ensuring a reliable and efficient power supply. To characterize the complicated fluctuation of wind power, numerous ensemble models with multiple base forecasters have been developed. However, most existing ensemble forecasting models contain several modeling stages, which increases the risk of error accumulation and inefficiency in model training. Moreover, the limited number of base forecasters results in forecasts with reduced diversity, thereby diminishing the performance of ensemble models. To address these challenges, MG-DS, a simple but efficient end-to-end ensemble learning model based on the Dempster-Shafer (DS) evidence theory, is proposed to unify base model learning and ensemble learning into a single process. It comprises an all-MLP-based nonlinear feature extraction module, a GRU and cross attention-based base forecast generation module, and a DS-based self-ensemble forecasting module with a DS-based magnifying glass to enhance the diversity of base forecasts. Further, a DS-based self-ensemble (DSSE) plugin is proposed to integrate the trained RNN-type and non-RNN-type base forecasters. Experiments on five wind power datasets show that MG-DS outperforms popular wind power forecasting models and ensemble techniques, and the effectiveness of the DSSE plugin is also validated in enhancing the performance of ensemble wind power forecasting.**

***Index Terms*—Wind power forecasting, end-to-end ensemble learning, DS-based magnifying glass, DS-based self-ensemble plugin.**

I. Introduction

WIND energy, as a clean and renewable energy source, is playing an increasingly important role in addressing the global energy crisis and climate change. According to the International Renewable Energy Agency (IRENA), global renewable power capacity had reached 3,870 GW at the end of 2023, marking an increase of 473 GW (+13.9%) during that year. Wind energy ranked second in capacity expansion, with a notable increase of 116 GW (+12.9%), following solar energy. With the rapid development of wind power technology and the continuous growth of installed capacity, wind power has become an essential component of modern power systems [1]. However, the intermittency and randomness of wind speed pose significant management and scheduling challenges for integrating wind power into the grid [2], [3]. Therefore, accurate wind power forecasting methods are crucial for enhancing the competitiveness of wind energy and the reliability of the grid.

Existing wind power forecasting methods can be broadly classified into two categories: statistical methods and artificial intelligence (AI)-based methods [4]. Statistical methods, including autoregressive (AR), autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA), and hidden Markov model (HMM), typically capture the linear fluctuations in wind power. Tang et al. [5] employed AR to extract linear patterns in wind power data, addressing the problem of model output insensitivity to input scale. Ahn et al. [6] proposed a practical short-term wind power forecasting method by incorporating exogenous variables into the ARIMA model. Li et al. [7] developed a novel HMM to improve the performance of wind power forecasting. However, due to the complex fluctuation characteristics of wind power, developing a purely linear model are insufficient for accurate forecasting.

Although statistical models can handle nonlinear relationship through appropriate preprocessing, transformations, or extensions, AI-based methods typically provide greater flexibility in capturing intricate patterns directly, often without the need for such extensive processing [8]. Therefore, many traditional machine learning methods and recent deep learning models have been used in wind power forecasting [9]. Machine learning methods include support vector machine (SVM), fuzzy logic, extreme learning machine (ELM), etc. Li et al. [10] introduced an enhanced SVM model optimized by the cuckoo search algorithm for short-term wind power forecasting. Liu and Wang [11] applied model-based transfer learning to construct a multi-layer ELM model, enhancing the accuracy of quantile wind power forecasting. Deep learning models primarily consist of convolutional neural network (CNN), recurrent neural network (RNN), long short-term memory (LSTM), temporal convolutional network (TCN), and other learning components [12]. Zhang et al. [13] combined CNN with geographic information system analysis technology for province-level total wind power forecasting. Wang et al. [14] proposed a wind power forecasting model based on an enhanced LSTM network to capture the nonlinearity between data variables and wind power. Gong et al. [15] utilized TCN to extract hidden temporal features from the data, followed by Informer to forecast wind power.

This work was supported in part by the National Natural Science Foundation of China under Grant 62376289, and in part by the Natural Science Foundation of Hunan Province, China under Grant 2024JJ4069. (Corresponding author: Yaohui Huang and Fan Zhang).

Yun Wang, Houhua Xu, Yaohui Huang, Fan Zhang, Hongbo Kou, and Runmin Zou are with the School of Automation, Central South University, Changsha, China (e-mail: wangyun15@tju.edu.cn; csuxuhouhua@csu.edu.cn, yaohuihuang@csu.edu.cn, zhangfan219@csu.edu.cn, kouhb23@ruc.edu.cn, rmzou@csu.edu.cn).

Qinghua Hu is with the College of Intelligence and Computing, Tianjin University, Tianjin, China (e-mail: huqinghua@tju.edu.cn).

Dipti Srinivasan is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore (e-mail: dipti@nus.edu.sg).

Wind power data exhibit intermittency and a high degree of randomness [16]. Therefore, a single model is not the best choice for wind power forecasting [17]. For this reason, numerous hybrid models have been developed aiming to improve the accuracy of wind power forecasts [18]. The most commonly used technique in hybrid modeling is ensemble learning, in which the most important consideration is to ensure diversity [19]. Specifically, in wind power forecasting, ensemble learning techniques are categorized into three groups based on the source of diversity: data diversity, forecaster diversity, and parameter diversity [20]. Ensemble methods based on data diversity introduce variability by training the base learner on different subsets of data [21]. Kim and Baek [22] combined wavelet transform with bootstrap to construct an ensemble model, which surpassed comparative forecasting methods. Many ensemble models focus on forecaster diversity [23], [24]. They employ multiple heterogeneous models to generate individual forecasting results, which are then combined to enhance forecasting accuracy [25]. In [26], three base forecasters, multi-layer perceptron network (MLP), improved CNN, and multi-layer LSTM, were combined using the stacking ensemble learning method to improve wind power forecasting performance. Dong et al. [27] selected several models with excellent forecasting ability as base forecasters, and aggregated the base forecasts using Hermite neural network with strong generalization ability. When the base forecasters are homogeneous, parametric diversity can be achieved by varying their weights or hyperparameters. Multiple forecasters, generated by perturbing parameters, were aggregated to achieve a precise and robust learner, which was proven to excel in terms of forecast accuracy and statistical tests [19].

The majority of existing ensemble models have some limitations. First, these methods achieve diversity primarily through two approaches: employing laborious data decomposition or training multiple base models [28]. As the required diversity increases, the number of model parameters and the computational time also increase, leading to a higher risk of overfitting and reduced scalability when dealing with large-scale datasets. Second, these methods commonly integrate diverse base forecasts using ensemble strategies such as voting [29], bagging [30], gradient boosting [31], and other adaptive ensemble strategies [23]. Wang el al. [32] and Tabassian et al. [33] treated the outputs of the base forecasters as 'evidence' and utilized the Dempster-Shafer (DS) evidence theory to enhance performance by effectively handling model uncertainty. Nevertheless, the almost processes of generating and merging base forecasts are handled separately. This separation increases training complexity and leads to error accumulation due to information loss between stages, ultimately resulting in suboptimal ensemble forecasts.

To address the limitations of two-stage ensemble learning, recent approaches have introduced a simple end-to-end ensemble learning framework based on the snapshot strategy [34] and its variants [35], [36]. Garipov et al. [35] proposed the Fast Geometric Ensembling (FGE) method, which employs a cyclical learning rate to generate diverse base forecasts. These forecasts are then directly combined through model averaging. Similarly, Izmailov et al. [36] introduced the Stochastic Weight Averaging (SWA) method to enhance generalization. This approach averages model weights sampled along the optimization trajectory of stochastic gradient descent (SGD) using either cyclical or constant learning rates and consolidates these averaged weights into a single model to generate the final forecasts. However, these methods rely on static weights within the ensemble function, limiting their ability to adapt to data with high variability or non-stationary characteristics, particularly in the task of wind power forecasting.

To address the above challenge, MG-DS, a novel end-to-end ensemble model based on the DS evidence theory [37], [38], is proposed for enhancing wind power forecasting. The model comprises three main components: an all-**M**LP-based nonlinear feature extraction module, a **G**RU and cross attention (CA)-based base forecast generation module, and a **DS**-based self-ensemble forecasting module. In module I, to enhance feature extraction efficiency, the all-MLP-based residual block is employed to extract intricate nonlinear features from the historical wind power data. In module II, the extracted nonlinear features and the temporal features extracted by GRU are fused by CA, and then three independent linear layers map the fused features into base forecasts. In module III, the diversity of base forecasts is enhanced using the DS evidence theory, and the final ensemble forecast is generated through a proposed adaptive dynamic ensemble, leveraging CNN and multi-head self-attention (MHSA) mechanism. This dynamic weighting strategy enables the model to assign context-dependent importance to each base forecast, thereby enhancing its adaptability to non-stationary patterns and high-variability data. Moreover, the module III is also designed as a DS-based self-ensemble (DSSE) plugin to combine several trained RNN-type or non-RNN-type base forecasters. In our proposed model, each module utilizes straightforward techniques to ensure simplicity and computational efficiency. This design serves as a baseline for end-to-end ensemble learning. The effectiveness of the proposed model and the DSSE plugin are evaluated on four wind power datasets. The main contributions of this study are as follows:

- A simple but efficient end-to-end ensemble framework is proposed for wind power forecasting. The framework unifies the process of base model learning and ensemble learning into a single process, thereby reducing error accumulation and improving modeling efficiency.
- With a limited number of base forecasters, a DS theory-based layer is designed and served as a magnifying glass to amplify the diversity of forecasts. This layer enhances forecasting capabilities by effectively fusing multiple forecasters, thus improving forecasting accuracy.
- An adaptive dynamic ensemble strategy is proposed to dynamically assign context-aware weights to the diverse base forecasts, thereby improving the model's adaptability to non-stationary and highly variable wind power data.
- The proposed self-ensemble strategy is extended to a useful plugin, enabling seamless fusion with a diverse range of trained forecasting models, encompassing both RNN-type and non-RNN-type variants, thereby amplifying the ensemble framework's adaptability and flexibility.

The rest of the paper is organized as follows: Section I describes the background and motivation; Section III presents the proposed model; Section IV illustrates the extended plugin; Section V provides a flowchart of the proposed model for wind power forecasting; Section VI presents forecasting results on four real-world datasets; Section VII offers further discussions, and Section VIII concludes the paper.

II. Background and Motivation

A. DS evidence theory

DS evidence theory, proposed by Dempster [37] and Shafer [38], is a powerful framework for handling uncertainty and making decisions in situations where evidence comes from multiple sources. A brief introduction of DS evidence theory is presented as follows.

Given the frame of discernment $\Theta = \{E_1, E_2, \cdots, E_n\}$, in which n elements in Θ are mutually exclusive and exhaustive, the power set 2^Θ, the set of all possible subsets of Θ, is expressed as

$$2^\Theta = \{\emptyset, \{E_1\}, \cdots, \{E_n\}, \cdots, \{E_1, \cdots, E_i\}, \cdots, \Theta\}, \quad (1)$$

where $\emptyset$ denotes the empty set, and $i \neq n$.

Within this frame of discernment, a basic probability assignment (BPA) function, also known as mass function, is a mapping function $m : 2^\Theta \rightarrow [0, 1]$, which satisfies the following two conditions:

$$\begin{cases} m(\emptyset) = 0 \\ \sum_{A \subseteq \Theta} m(A) = 1 \end{cases} \quad (2)$$

where A is a subset of Θ, and $m(A)$ represents the degree of belief that the evidence provides for A.

Supposing m_1 and m_2 are two BPA functions induced by two independent items of evidence. These pieces of evidence can be fused using Dempster's combinational rule,

$$m(A) = (m_1 \oplus m_2)(A) = \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{1 - \sum_{B \cap C = \emptyset} m_1(B) m_2(C)}. \quad (3)$$

For more improved combination rules, see [32].

B. Ensemble learning

Compared to a single model, ensemble learning leverages the strengths of multiple models to significantly enhance the handling of complex tasks and improve forecasting accuracy. To enhance the wind power forecasting accuracy, some ensemble models have been developed that utilize the initial forecasts of base models as inputs, as shown in **Fig. 1 (a)**.

Given the feature x and the target y, the ensemble model with K base forecasters is expressed as

$$y = g([\hat{f}_1(x), \cdots, \hat{f}_K(x)]) + e', \quad (4)$$

where $g(\cdot)$ is the ensemble function, e' is the error term, and $\hat{f}_i(\cdot)$ is the estimation for the ith base forecasting model,

$$y = f_i(x) + e'', \quad (5)$$

where e'' is the error term.

From (4), there are two main steps in the two-stage ensemble learning framework to generate the final forecasts: first, training K base forecasting models on historical data to obtain $\{\hat{f}_1(\cdot), \cdots, \hat{f}_K(\cdot)\}$ and generating K base forecasts, which serve as inputs for the second step, where the ensemble model is trained using these initial forecasts to obtain $\hat{g}(\cdot)$, which is then used to generate the final ensemble forecast.

In addition, some conventional end-to-end ensemble models streamline the training process by jointly optimizing the base forecasting models and the ensemble function, as illustrated in **Fig. 1 (b)**. Given the feature x and the target y, the end-to-end framework simultaneously trains the base forecasting models $\{f_1(\cdot), \ldots, f_K(\cdot)\}$ and the ensemble model $g(\cdot)$ using historical data. The objective is to minimize the overall loss by directly integrating the ensemble function into the optimization process, enabling the framework to produce the final forecasts in a single training phase. However, conventional end-to-end ensemble models often rely on static weights in the ensemble function $g(\cdot)$, which can restrict their ability to adapt to data with high variability or non-stationary characteristics.

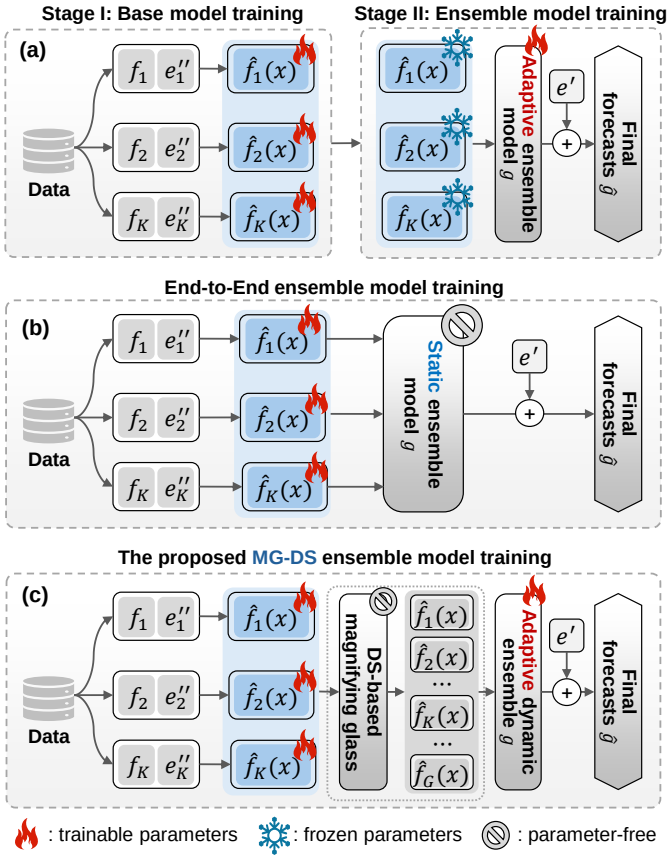


Fig. 1. The ensemble learning frameworks. (a) Two-stage ensemble learning framework. (b) Conventional end-to-end ensemble learning framework. (c) The proposed MG-DS ensemble learning framework.

C. Motivation

According to the ensemble learning process, there are several main challenges in generating high-quality ensemble forecasts:

- The two-stage ensemble learning approach relies on base forecasting results from the first stage to directly influence

the training of the ensemble model in the next stage [23], [39]. This dependency can lead to error accumulation, where inaccuracies in the base forecasters propagate through the process and reduce the performance of the final ensemble [24], [40].
- The limitation in the number of base forecasters restricts the ensemble model's ability to capture sufficient diversity, thereby reducing its generalization capability and overall performance [25].
- The use of an increased number of base forecasters can enhance diversity but introduces higher computational costs and risks of high similarity between forecasters [41]. This similarity may cause forecasters to reinforce each other's errors, ultimately decreasing performance [42].
- The conventional end-to-end frameworks combine forecasts using static weights, which limits their flexibility and adaptability to the dynamic nature of real-world scenarios [23].

Considering these challenges, this study proposes a novel end-to-end ensemble learning framework MG-DS, as depicted in **Fig. 1 (c)**. In contrast to the traditional two-stage ensemble learning framework, the proposed approach trains the base forecasting models and the ensemble model simultaneously. This joint training strategy minimizes error propagation by removing dependencies between stages. To address the issue of limited base forecasters, the framework incorporates a DS-based magnifying glass module based on DS evidence theory. This parameter-free approach enhances forecast diversity without increasing the number of base models. Furthermore, MG-DS employs an adaptive dynamic ensemble that addresses the limitations of conventional end-to-end frameworks by dynamically adapting to changing data patterns, thereby improving the ensemble model's forecasting capabilities. Overall, MG-DS provides a flexible and adaptive solution for ensemble learning, improving forecasting performance and computational efficiency in dynamic real-world scenarios.

III. DS-BASED END-TO-END ENSEMBLE MODEL

Within the proposed framework in **Fig. 1 (c)**, this study designed a novel DS-based end-to-end ensemble model MG-DS. Its detailed structure is shown in **Fig. 2**. The proposed model can be divided into three main modules.

A. Module I: All-MLP-based nonlinear feature extraction

Basic linear models have demonstrated simplicity and effectiveness in recent studies on time series forecasting [43]. In this study, an MLP-based encoder similar to that in [43] is designed to extract nonlinear features from historical wind power data. As depicted in **Fig. 2**, all-MLP-based residual block is used as the basic layer for feature extraction.

Given L-dimensional historical wind power series $\mathbf{X}_t = [x_{t-L+1}, x_{t-L+2}, \cdots, x_t] \in \mathbb{R}^L$ at time t, it is initially encoded as a matrix $\mathbf{X}_{in} \in \mathbb{R}^{L\times d}$ through a linear embedding layer. After that, the embedded data is passed through two residual blocks to extract nonlinear features from the input data. Specifically, $\mathbf{X}_{in}$ is initially passed through a linear layer with weight $\mathbf{W}_1$, and bias $\mathbf{b}_1$, and ReLU activation function,

$$\mathbf{X}' = \texttt{ReLU}(\mathbf{W}_1 * \mathbf{X}_{in} + \mathbf{b}_1), \tag{6}$$

where $*$ denotes the matrix multiplication operation.

Subsequent to the ReLU activation, the output undergoes a transformation through a second linear layer, employing a linear activation function. And, dropout is applied to the linear layer with weight $\mathbf{W}_2$, and bias $\mathbf{b}_2$ to obtain the output $\mathbf{X}^{''}$,

$$\mathbf{X}^{''} = \texttt{Dropout}\big(\texttt{Linear}(\mathbf{W}_2 * \mathbf{X}^{'} + \mathbf{b}_2)\big). \tag{7}$$

A skip connection is then introduced, facilitating a direct flow of information from the input of the module to the output, enhancing gradient propagation and preventing the vanishing gradient problem. Lastly, a layer normalization is employed to normalize the output, which stabilizes the learning process and accelerates the convergence of the training. The obtained feature matrix can be expressed as

$$\mathbf{H}_{en} = \texttt{LayerNorm}\big(\mathbf{X}^{''} + \texttt{Linear}(\mathbf{W}_3 * \mathbf{X}_{in} + \mathbf{b}_3)\big) \in \mathbb{R}^{L\times d}, \tag{8}$$

where $\mathbf{W}_3$ and $\mathbf{b}_3$ are the weights and bias of a linear layer.

B. Module II: GRU and CA-based base forecast generation

In module II, a GRU and CA-based module is designed for generating base forecasts. As shown in **Fig. 2**, the input historical data $\mathbf{X}_t \in \mathbb{R}^L$ at time t is passed through three GRU layers to extract the complex temporal features $\mathbf{H}_{tem}$,

$$\mathbf{H}_{tem} = \texttt{GRU}(\mathbf{X}_t) \in \mathbb{R}^{L\times d}. \tag{9}$$

To account for the nonlinear feature $\mathbf{H}_{en} \in \mathbb{R}^{L\times d}$ extracted by the encoder layer, the CA mechanism is used to fuse it with the extracted complex temporal feature $\mathbf{H}_{tem}$. The CA is computed in the same way as the self-attention mechanism, with the difference that the query vectors, key vectors, and value vectors are from different sources. Specifically, the query vector is $\mathbf{H}_{tem}$, while both the key and value vectors are $\mathbf{H}_{en}$. The fusion feature $\mathbf{H}_{ca} \in \mathbb{R}^{L\times d}$, which is obtained by CA, is denoted as

$$\mathbf{H}_{ca} = \texttt{softmax}\left(\frac{\mathbf{H}_{tem}\mathbf{H}_{en}^{\top}}{\sqrt{d}}\right)\mathbf{H}_{en} \in \mathbb{R}^{L\times d}. \tag{10}$$

The extracted $\mathbf{H}_{ca}$ is concatenated with the temporal feature $\mathbf{H}_{tem}$ to obtain the final feature $\mathbf{H}_{de}$, which is expressed as

$$\mathbf{H}_{de} = \texttt{Concat}(\mathbf{H}_{ca}, \mathbf{H}_{tem}) \in \mathbb{R}^{L\times 2d}. \tag{11}$$

Subsequently, K independent linear layers map the feature $\mathbf{H}_{de}$ into the base forecast matrix $\mathbf{R}$,

$$\mathbf{R} = [\mathbf{R}^1, \cdots, \mathbf{R}^K] \in \mathbb{R}^{L\times K}, \tag{12}$$

$$\mathbf{R}^j = \mathbf{W}^j * \mathbf{H}_{de} + \mathbf{b}^j \in \mathbb{R}^{L\times 1}, \tag{13}$$

where $\mathbf{W}^j$ and $\mathbf{b}^j$ denote the weight and bias of the jth linear layer. Considering the limited number of base forecasters in real applications and model efficiency, K is set to 3 in this study. Due to the inherent characteristics of GRU, where information is transmitted strictly in a backward manner along the time step, the forecasts of the jth linear layer at time t can be expressed as $\mathbf{R}^j = [\hat{y}^j_{t-L+2}, \hat{y}^j_{t-L+3}, \cdots, \hat{y}^j_{t+1}]$ with the input data $\mathbf{X}_t$.

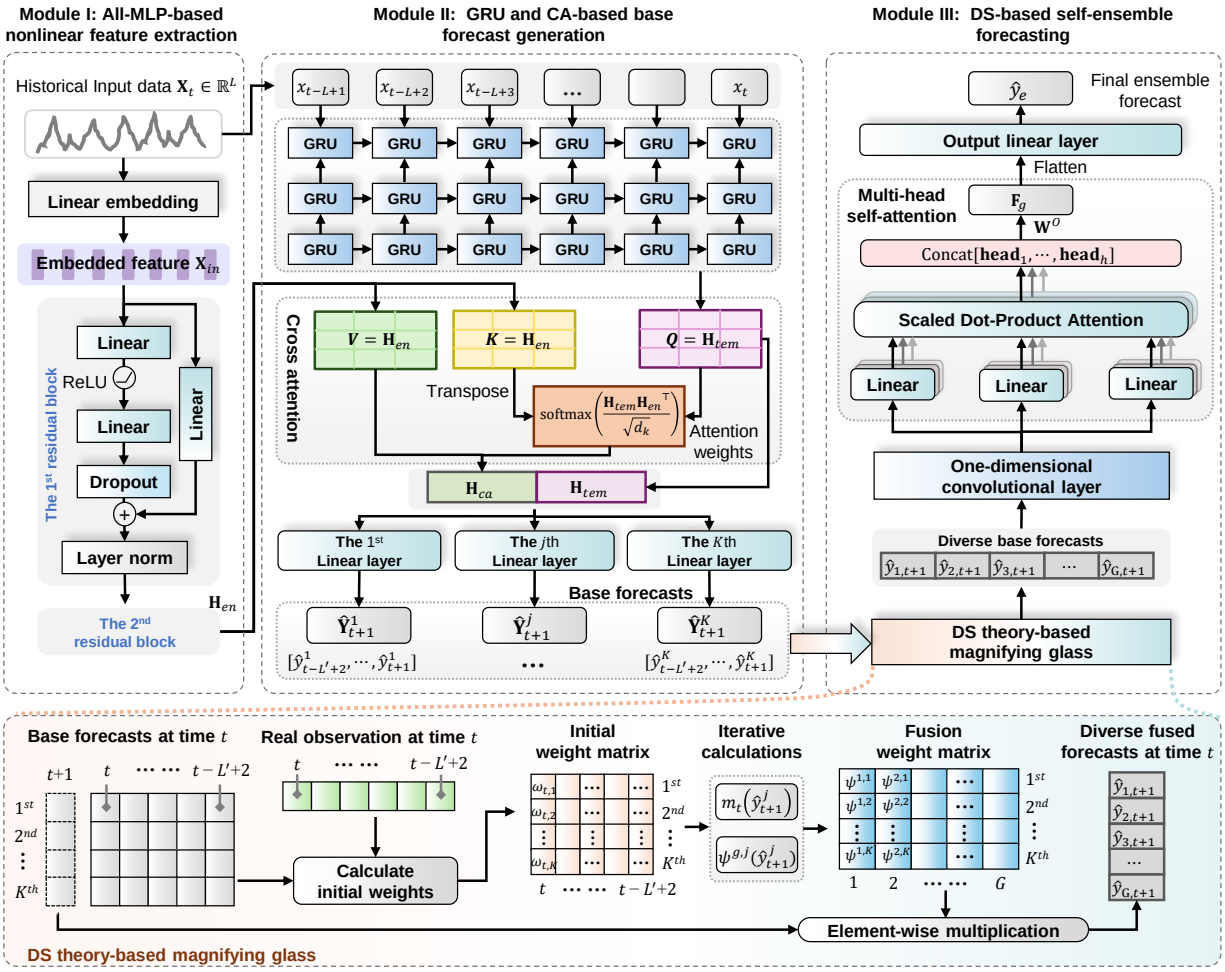


Fig. 2. Structure of the proposed model.

C. Module III: DS-based self-ensemble forecasting

The diversity of base forecasts plays a crucial role in ensemble learning. Common methods for achieving diverse base forecasts involve using different models or feature sets, often necessitating multi-stage training. However, this approach introduces complexity to both training and forecasting processes, and may result in error accumulation across different stages. The module III focuses on how to enhance the diversity of the base forecasts with the limited number of base forecasters within an end-to-end framework. Therefore, a DS evidence theory-based self-ensemble forecasting module, as shown in **Fig. 2**, is developed with two primary components: a DS-based magnifying glass and a CNN and MHSA-based forecaster.

1) DS-based magnifying glass

The role of DS-based magnifying glass is to generate more diverse forecasts when there are limited numbers of base forecasts, as shown in **Fig. 2**. The key idea is to generate different fusion weights based on the DS evidence theory. At first, it is necessary to extract the initial weights. In this case, assume there are K base forecasters, where K is a small value. The historical wind power data with length L construct the input data $\mathbf{X}_t = [y_{t-L+1}, y_{t-L+2}, \cdots, y_t]$ at time t, and $L^{'}$ wind power forecasts before the time $t+1$ (i.e., $\hat{\mathbf{Y}}_{t+1}^j = [\hat{y}_{t-L'+2}^j, \hat{y}_{t-L'+3}^j, \cdots, \hat{y}_{t+1}^j]$, $j = 1, \cdots, K$) are generated by the jth base forecaster according to the characteristics of GRU. The symbols y_t and $\hat{y}_t^j$ denote the actual wind power and the corresponding power forecast of the jth base forecaster at time t, respectively. Considering the rule that using historical data to forecast future wind power, there is a constraint that $L^{'} \leq L$.

For L' wind power forecasts of the jth base forecaster at time t, except for the forecast $\hat{y}_{t+1}^j$, there are $L'-1$ forecasts $[\hat{y}_{t-L'+2}^j, \cdots, \hat{y}_t^j]$ that have the known targets, i.e., $[y_{t-L'+2}, \cdots, y_t]$. Therefore, this study actually focuses on one-step-ahead wind power forecasting by considering the previous forecasts. In general, a more accurate forecasting model will be assigned with a large weight. Therefore, the initial weights for the jth base forecaster at time t can be defined as

$$w_{t,j} = \frac{1}{\text{Error}(y_t, \hat{y}_t^j) \cdot \sum_{j=1}^{K} 1/\text{Error}(y_t, \hat{y}_t^j)}, \qquad (14)$$

where the function $\text{Error}(\cdot,\cdot)$ is an error measure that describes the difference between the forecast and the corresponding observation. Many types of $\text{Error}(\cdot,\cdot)$ can be defined, such

as mean squared error and mean absolute error (MAE). In this case, $\text{Error}(y_t, \hat{y}_t^j) = |y_t - \hat{y}_t^j|$ is used for its robustness.

In this study, the frame of discernment is defined as $\Theta = \{\hat{y}_{t+1}^1, \hat{y}_{t+1}^2, \ldots, \hat{y}_{t+1}^K\}$, which contains K base forecasts. According to the model performance at different historical time, different model weights can be obtained through (14). Considering the similarity of the BPA function value and model weight, the tth BPA function value of the forecast $\hat{y}_{t+1}^j$ generated by the jth base forecaster is assigned with its weight calculated based on all K forecasts at time t,

$$m_t(\hat{y}_{t+1}^j) = w_{t,j}, \quad j = 1, \cdots, K. \tag{15}$$

According to the Dempster's combinational rule, the evidence from the forecasts at time t and $t-1$ can be fused by the following calculation,

$$\psi^1(\hat{y}_{t+1}^j) = \frac{m_{t-1}(\hat{y}_{t+1}^j) \cdot m_t(\hat{y}_{t+1}^j)}{1 - \sum_{j \neq j'} m_{t-1}(\hat{y}_{t+1}^j) \cdot m_t(\hat{y}_{t+1}^{j'})}. \tag{16}$$

where $\psi^1(\hat{y}_{t+1}^j)$ denotes the value of BPA function after the first round of evidence fusion. When considering the evidence from the forecasts at time $t-2$, the value of BPA function after the second round of evidence fusion is given by

$$\psi^2(\hat{y}_{t+1}^j) = \frac{m_{t-2}(\hat{y}_{t+1}^j) \cdot \psi^1(\hat{y}_{t+1}^j)}{1 - \sum_{j \neq j'} m_{t-2}(\hat{y}_{t+1}^j) \cdot \psi^1(\hat{y}_{t+1}^j)}. \tag{17}$$

Assuming there are G fusion rounds, the BPA function value $\psi^g(\hat{y}_{t+1}^j)$ for $\hat{y}_{t+1}^j$ in the gth ($g \geq 2$) fusion round is computed with a cycle manner based on the BPA function values at time $t-g$ and those obtained after the $(g-1)$th round of evidence fusion,

$$\psi^g(\hat{y}_{t+1}^j) = \frac{\sum_{A \cap A' = \{\hat{y}_{t+1}^j\}} m_{t-g}(A) \cdot \psi^{g-1}(A')}{1 - \sum_{A \cap A' = \emptyset} m_{t-g}(A) \cdot \psi^{g-1}(A')}, \tag{18}$$

where A and A' are random elements in Θ.

Different fusion rounds produce varying BPA function values, which are considered as fusion weights of K base forecasts. Based on these fusion weights, at time t, the fused forecast $\hat{y}_{g,t+1}$ for time $t+1$ at the gth evidence fusion round can be computed as

$$\hat{y}_{g,t+1} = \sum_{j=1}^{K} \psi^g(\hat{y}_{t+1}^j) \cdot \hat{y}_{t+1}^j, g = 1, \cdots, G. \tag{19}$$

Therefore, G base forecasts for time $t+1$ can be obtained,

$$\hat{\boldsymbol{y}}_{t+1}^{DS} = [\hat{y}_{1,t+1}, \hat{y}_{2,t+1}, \ldots, \hat{y}_{G,t+1}] \in \mathbb{R}^{1 \times G}. \tag{20}$$

For any arbitrary $G > K$, the above computations serve as a magnifying glass that extends K original base forecasts into G base forecasts, thus enhancing the diversity of base forecasts.

2) CNN and MHSA-based forecaster

Having obtained the expanded forecasts $\hat{\boldsymbol{y}}_{t+1}^s$, the subsequent challenge is to determine the optimal method for ensembling them to achieve improved final forecasts. Therefore, as shown in **Fig. 2**, an adaptive dynamic ensemble learning strategy that leverages CNN and MHSA mechanism is designed to generate the final ensemble forecasts.

First, three one-dimensional convolutional layers with multiple kernels of size r are employed to extract the high-dimensional features $\mathbf{F}_c$ from the diverse base forecast $\hat{\boldsymbol{y}}_{t+1}^s$, which can be expressed as the following formula:

$$\mathbf{F}_c = \left[C_{r_3}^3 \otimes \left(C_{r_2}^2 \otimes (C_{r_1}^1 \otimes \hat{\boldsymbol{y}}_{t+1}^s)\right)\right]^\top \in \mathbb{R}^{G \times nk}, \tag{21}$$

where nk is the number of kernels, $C_{r_i}^i$ denotes the ith one-dimensional convolutional layer with kernel size r_i, and $\otimes$ represents the convolutional operation.

The features $\mathbf{F}_c$ are then filtered and fused by MHSA to obtain the global features $\mathbf{F}_g$,

$$\mathbf{F}_g = \texttt{Concat}(\mathbf{head}_1, \ldots, \mathbf{head}_h)\mathbf{W}^O \in \mathbb{R}^{G \times nk}, \tag{22}$$

$$\mathbf{head}_i = \texttt{softmax}\left(\frac{(\mathbf{F}_c\mathbf{W}_i^Q)(\mathbf{F}_c\mathbf{W}_i^K)^\top}{\sqrt{G}}\right)(\mathbf{F}_c\mathbf{W}_i^V), \tag{23}$$

where $\mathbf{head}_i \in \mathbb{R}^{G \times (nk/h)}$ represents the ith attention head, $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V \in \mathbb{R}^{nk \times (nk/h)}$, and $\mathbf{W}^O \in \mathbb{R}^{nk \times nk}$ are the projection matrices.

Finally, the global features $\mathbf{F}_g$ are flattened and then mapped through a linear layer with weight $\mathbf{W}_e$ and bias $\mathbf{b}_e$ to generate the final ensemble forecast $\hat{y}_e$,

$$\hat{y}_e = \texttt{Linear}(\mathbf{W}_e * \texttt{Flatten}(\mathbf{F}_g) + \mathbf{b}_e). \tag{24}$$

This dynamic ensemble is characterized by the fact that the implicit ensemble weights adapt to the changing diverse forecasts, thereby ensuring the adaptive nature of the ensemble.

IV. Model Extension

The DS-based self-ensemble strategy in module III is not only suitable for constructing an end-to-end ensemble model, but is also regarded as an independent plugin for ensembling a limited number of trained base forecasters. The objective of this section is to explore how the proposed plugin can seamlessly ensemble with different model architectures, encompassing both RNN variations and models outside the RNN paradigm.

A. The DSSE plugin

The module III in the proposed model is regarded as an independent plugin, called the DSSE plugin. **Algorithm 1** outlines the entire computation process. With the limited number of base forecasts fed into **Algorithm 1**, the DSSE plugin can generate the final ensemble forecast.

B. The DSSE plugin for the trained RNN-type models

RNN-type models are popular time series forecasting models and have been widely used for wind power forecasting. When there are K trained RNN-type base wind power forecasters, with the input $\mathbf{X}_t$ at time t, the historical hidden states can be obtained owing to the nature of RNN-type models. These states are used to generate base forecasts $\hat{\mathbf{Y}}_{t+1}^j$. In this case, the DSSE plugin can be directly used to ensemble the K trained base models and improve the wind power forecasting performance.

Algorithm 1 The DSSE Plugin.

Input: K base forecasts $\{\hat{\mathbf{Y}}^1_{t+1}, \cdots, \hat{\mathbf{Y}}^K_{t+1}\}$, and the historical time series data $\mathbf{X}_t$.

Output: Final ensemble forecast $\hat{y}_e \in \mathbb{R}$.

1) Calculate the forecasting error based on Error$(y_t, \hat{y}^j_t)$;
2) Obtain the initial weight matrix $\boldsymbol{w} = \{w_{t,j}\}$ based on (14), where $\boldsymbol{w} \in \mathbb{R}^{(L-1)\times K}$;
3) Obtain diverse ensemble weights $\psi^g(\hat{y}^j_{t+1})$ based on (16), (17), and (18) under different fusion steps;
4) Generate diverse base forecasts based on (19);
5) Construct the feature map $\mathbf{F}_c$ using three one-dimensional convolutional layers based on (21);
6) Get the global features $\mathbf{F}_g$ by (22) and (23);
7) Obtain the final ensemble forecast $\hat{y}_e \in \mathbb{R}$ with (24).

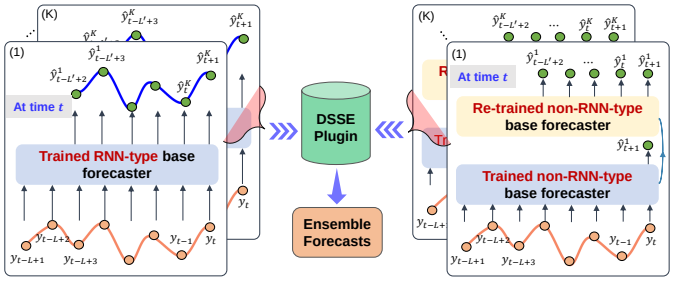


Fig. 3. Use of DSSE for different types of base forecasters.

C. The DSSE plugin for the trained non-RNN-type models

For trained non-RNN-type base wind power forecasting models, an adjustment and retraining process is necessary, as these models generally do not produce historical hidden states. The framework of DSSE applied to non-RNN-type base models is shown in **Fig. 3**. First, reconstruct the output of the jth base model so that the output contains not only the forecasts at time $t+1$, i.e., $\{\hat{y}^j_{t+1}\}$, but also the historical forecasts at time t, $\{\hat{y}^j_{t-L'+2}, \hat{y}^j_{t-L'+3}, \ldots, \hat{y}^j_t\}$. Then, train the DSSE plugin with the constructed data form. The trained plugin expands the forecast diversity and finally produces ensemble forecasts through ensemble learning.

V. Flowchart of Wind Power Forecasting with the Proposed Model and DSSE Plugin

With the proposed model and plugin, wind power forecasting can be realized by the following steps.

Step 1: Data acquisition and pre-processing. Some wind power datasets publicly available from open-source websites were acquired for evaluating the forecasting capabilities of different models. Each dataset was divided into distinct non-overlapping subsets: the training set, validation set, and test set. Moreover, to avoid the impact of numerical differences in input features and ensure model stability and accuracy, all data were normalized to the range [0,1] by:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}, \tag{25}$$

where x_{norm} is the normalized data, x is the raw data, and x_{max}, x_{min} are the maximum and minimum values in the training set.

Step 2: Input length selection. In time series forecasting, selecting the appropriate length of the input is essential. In this study, Spearman correlation coefficient (SCC) [44] is used to characterize the degree of correlation between various moments of the historical data and the target. With this nonparametric approach, we are able to capture the latent linear and non-linear relationships. Subsequently, a threshold value for the correlation coefficient is set to filter out the historical moments that are significantly correlated with the forecast target, thus determining the appropriate input history length.

Step 3: Model training and hyperparameter determination. Model training and determination of hyperparameters need to be carefully tuned to achieve optimal performance. During the training process, the hyperparameters of the model are adjusted by evaluating the performance of model on the validation set. This is an iterative optimization process that gradually approaches the best configuration by trying different parameter combinations. To avoid the overfitting problem, the early stopping mechanism is incorporated. The model's performance on the validation set is monitored during training to prevent overfitting on the training set, ensuring the model's reliability and adaptability.

In this study, when training the proposed self-ensemble model MG-DS, all the parameters in the model are optimized simultaneously by continuously adjusting its hyperparameters, and the optimal ensemble model is finally obtained. When DSSE is used to ensemble the trained base models, the parameters of the base models are kept frozen during the training process, while only the DSSE parameters are optimized.

Step 4: Wind power forecasting and evaluation. After the model is trained and optimized, it is used to make forecasts on the test set. The forecast $\hat{y}_{norm}$ obtained from the model is denormalized back to their original scale $\hat{y}$ using the inverse of the normalization formula applied in Step 1:

$$\hat{y} = \hat{y}_{norm} \cdot (x_{max} - x_{min}) + x_{min}. \tag{26}$$

Meanwhile, various evaluation metrics are used to evaluate the forecasting performance of the model.

Through the above steps, a tuned MG-DS for wind power forecasting have been obtained. This model has good performance and can provide reliable forecasts in practical applications, providing strong support for decision-making and planning in the field of wind power.

VI. Case Study

In this part, case studies are conducted to assess the effectiveness of the proposed forecasting model and plugin. The details are presented below.

A. Dataset description

Five real-world wind power datasets from Belgium, Finland, Austria, Denmark, and Bulgaria were utilized to demonstrate the effectiveness of the proposed model. These datasets were obtained from open-source platforms (https://data.open-power-system-data.org/time_series/). For simplicity, these datasets are referred to as Dataset A, B, C, D, and

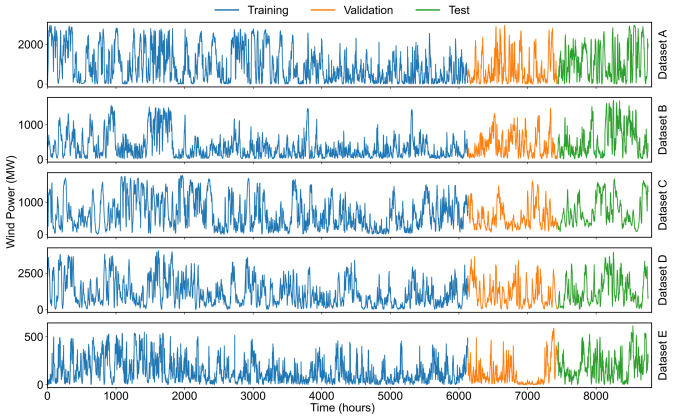


Fig. 4. The visualization of five wind power datasets used in this study.

E, respectively. Each dataset consists of hourly onshore wind power generation time series covering the entire year of 2019. Datasets A–D represent national-level, aggregated wind power generation from Belgium, Finland, Austria, and Denmark, with actual maximum hourly outputs exceeding 1000 MW during the study period. Dataset E provides corresponding national-level data for Bulgaria, where the actual maximum hourly generation in 2019 was below 1000 MW. **Table I** presents the statistical properties of all datasets. For each dataset, 70% of the data was allocated to the training set for model training, 15% to the validation set for hyperparameter selection, and the remaining 15% to the test set for evaluating the model performance. **Fig. 4** provides a comprehensive visualization of the time series data for all five datasets, illustrating the training, validation, and test splits.

B. Benchmark models

Two types of benchmark models were implemented to test the effectiveness of the proposed model. The first type comprises two traditional statistical models (the persistence model and AR). The second type consists of 15 AI-based models, including two traditional machine learning models (random forest regressor (RF) and BPNN), five popular deep learning models (CNN, LSTM, GRU, Mamba [45], and DLinear [46]), two hybrid models (CNN-GRU and MG), and six ensemble models (MG-V, MG-Ba, MG-Bo, MG-S, MG-FGE, MG-SWA). CNN-GRU is a sequential combination of CNN and GRU [47]. MG is a variant of the proposed model, consisting only of the first two modules, with a single linear layer to generate forecast in the second module. MG-V, MG-Ba, and MG-Bo are two-stage ensemble models that integrate multiple MGs using voting [29], bagging [30], and gradient boosting [31] ensemble strategies, respectively. MG-S, MG-FGE, and MG-SWA are end-to-end ensemble models that combine MGs using snapshot ensemble [34], FGE, and SWA, respectively.

For all forecasting models, the number of hidden neurons is selected from $\{32, 64, 128\}$, including those in the linear layer, recurrent neural network, and convolutional neural network. The size of the convolutional kernels is chosen from $\{3, 5\}$, and the number of attention heads is selected from $\{4, 8\}$. A grid search strategy is employed to evaluate all hyperparameter combinations and determine the optimal configuration based on model performance on the validation set. In the proposed MG method, the convolutional layer is set to three, and the size of convolutional kernels and the number of hidden neurons are searched independently for each layer. Similar restricted search ranges have been employed in previous studies [3], [12], since they offer a practical balance between feasibility and model performance. For model training, MAE is used as the loss function in this study.

TABLE I
STATISTICAL INFORMATION OF ALL DATASETS.

Dataset	Minimum	Maximum	Mean	Variance	Skewness	Kurtosis
A	4.0000	2967.0000	909.3990	822.5581	0.7739	-0.6303
B	8.3300	1701.0500	383.7908	354.8784	1.3847	1.2760
C	1.3000	1832.6000	647.9935	458.6851	0.7048	-0.5638
D	8.8300	4046.3300	1177.5476	895.9071	0.8554	-0.1297
E	0.0000	613.0000	142.8337	127.3521	1.1351	0.5611

C. Evaluation metrics

This study employed MAE, Root Mean Square Error (RMSE), determination coefficient (R^2), Weighted Mean Absolute Percentage Error (WMAPE), and Index of Agreement (IA) to assess the performance of various forecasting models. Their calculations are given by:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|, \tag{27}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, \tag{28}$$

$$\text{R}^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}, \tag{29}$$

$$\text{WMAPE} = \frac{\sum_{i=1}^{N}|y_i - \hat{y}_i|}{\sum_{i=1}^{N} y_i}, \tag{30}$$

$$\text{IA} = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(|y_i - \bar{y}| + |\hat{y}_i - \bar{y}|)^2}, \tag{31}$$

where N is the number of test samples, $y_i, \hat{y}_i$ are the ith actual wind power and the corresponding forecast, respectively, and $\bar{y}$ is the mean of all actual wind power data in the test set.

D. Optimal input determination

Before training various forecasting models, it is essential to determine the input length L (i.e., the lag order), which significantly impacts the model performance. Selecting suitable inputs is crucial for achieving accurate forecasts. In this study, the SCC was used to calculate the correlation between the target data and the historical data for different lag orders. **Fig. 5** presents the sensitivity analysis results for SCC threshold α. The analysis indicates that selecting $\alpha = 0.3$ results in relatively better performance in most cases. Consequently, for the hourly wind power datasets, historical wind power data with SCC values exceeding the threshold $\alpha = 0.3$ are identified as optimal inputs. The calculated SCC values for

TABLE II
WIND POWER FORECASTING RESULTS OF THE PROPOSED MODEL AND BENCHMARK MODELS.

Dataset	Metric	Statistical models		Machine learning models		Deep learning models					Hybrid models		Ensemble models						Ours
		Persistence	AR	RF	BPNN	CNN	LSTM	GRU	Mamba	DLinear	CNN-GRU	MG	MG-V	MG-Ba	MG-Bo	MG-S	MG-FGE	MG-SWA	MG-DS
Dataset A	MAE	121.9789	113.4870	107.5396	103.4521	102.0222	101.6225	101.0491	102.5252	100.6902	100.1901	100.1002	99.5051	99.7677	99.7761	99.7317	99.7446	99.7618	**98.7225**
	RMSE	175.4558	166.4251	151.7906	150.2635	144.8496	145.0784	145.4119	145.1253	143.3260	144.7297	144.6745	145.2745	145.6265	143.9988	143.8889	143.2885	143.8280	**141.9168**
	WMAPE	10.6007	9.8627	9.3459	8.9906	8.8664	8.8316	8.7818	9.1118	8.9488	8.7646	8.7617	8.6476	8.6704	8.6712	8.6635	8.8647	8.8662	**8.5796**
	R^2	0.9571	0.9614	0.9679	0.9686	0.9708	0.9707	0.9706	0.9684	0.9692	0.9707	0.9709	0.9706	0.9705	0.9711	0.9713	0.9692	0.9689	**0.9720**
	IA	0.9892	0.9904	0.9917	0.9921	0.9925	0.9925	0.9925	0.9920	0.9921	0.9925	0.9926	0.9926	0.9926	0.9926	0.9927	0.9922	0.9922	**0.9929**
	DM	9.62	6.13	5.11	4.53	3.50	4.81	3.05	3.88	3.87	3.89	3.84	1.20	1.48	1.87	0.43	1.44	1.07	-
	Sig. Level	***	***	***	***	***	***	***	***	***	***	***	None	None	*	None	None	None	-
Dataset B	MAE	59.0539	55.3506	55.5472	53.4256	52.2635	52.0700	51.6024	52.8027	50.9497	50.8146	50.6367	50.6392	50.5881	50.3078	50.4523	50.6685	50.0709	**49.8744**
	RMSE	82.2794	76.4319	77.6094	74.2358	73.0566	72.7470	70.9291	73.1811	70.6145	71.3779	70.7153	71.1787	71.0833	70.5442	70.8783	71.1105	70.2938	**69.8864**
	WMAPE	10.8072	10.1295	10.1655	9.7772	9.5645	9.5291	9.4435	9.6632	9.3241	9.2994	9.2668	9.2673	9.2579	9.2066	9.2331	9.2726	9.1633	**9.1273**
	R^2	0.9648	0.9696	0.9687	0.9713	0.9722	0.9725	0.9738	0.9721	0.9741	0.9735	0.9740	0.9736	0.9737	0.9741	0.9739	0.9737	0.9743	**0.9746**
	IA	0.9911	0.9924	0.9918	0.9927	0.9929	0.9929	0.9932	0.9929	0.9934	0.9934	0.9933	0.9934	0.9934	0.9934	0.9934	0.9934	0.9935	**0.9935**
	DM	9.3884	6.3156	6.9370	5.8263	4.8876	4.8080	4.6169	5.3646	2.8106	2.6775	2.9871	2.8554	2.9211	2.4459	2.2314	2.4448	0.8582	-
	Sig. Level	***	***	***	***	***	***	***	***	***	***	***	***	***	**	**	**	None	-
Dataset C	MAE	33.2959	29.6444	28.6768	28.6998	28.1916	27.9465	27.4633	29.2338	27.8053	27.3726	27.3806	27.3440	27.3495	27.4728	27.0755	27.1892	27.3132	**26.8003**
	RMSE	46.9291	40.9595	39.6003	40.0642	39.3941	38.8288	38.1677	39.9130	37.9404	37.9362	37.8394	37.7334	37.7459	37.8469	37.6892	38.1774	37.8838	**37.3362**
	WMAPE	4.5755	4.0737	3.9408	3.9439	3.8741	3.8404	3.7740	4.0173	3.8210	3.7616	3.7627	3.7552	3.7569	3.7753	3.7207	3.7364	3.7534	**3.6829**
	R^2	0.9885	0.9912	0.9918	0.9916	0.9919	0.9921	0.9924	0.9917	0.9925	0.9925	0.9925	0.9926	0.9926	0.9925	0.9926	0.9924	0.9925	**0.9927**
	IA	0.9971	0.9978	0.9979	0.9979	0.9979	0.9980	0.9981	0.9979	0.9981	0.9981	0.9981	0.9981	0.9981	0.9981	0.9981	0.9981	0.9981	**0.9982**
	DM	8.6381	5.6456	4.0946	3.6759	2.8416	4.6444	3.5489	3.6564	5.6720	3.7759	4.0603	2.1098	0.9273	4.0497	0.3701	0.9299	3.5728	-
	Sig. Level	***	***	***	***	***	***	***	***	***	***	***	**	None	***	None	None	***	-
Dataset D	MAE	99.6336	80.4111	79.6030	75.9627	75.2144	73.7632	74.1467	79.2521	75.0273	72.8824	72.1232	73.6171	73.5564	73.5361	72.6102	71.8848	71.8252	**71.5932**
	RMSE	139.6754	123.7795	118.7350	117.9710	117.2927	117.4935	116.9615	121.8257	117.8382	115.8237	114.8114	116.1170	116.1977	115.2185	115.3398	113.3581	115.0144	**112.5549**
	WMAPE	7.5160	6.0660	6.0050	5.7304	5.6739	5.5645	5.5934	5.9785	5.6598	5.4980	5.4407	5.5534	5.5489	5.5473	5.4775	5.4228	5.4183	**5.4008**
	R^2	0.9773	0.9821	0.9836	0.9838	0.9840	0.9839	0.9841	0.9827	0.9838	0.9844	0.9846	0.9843	0.9843	0.9845	0.9845	0.9850	0.9846	**0.9852**
	IA	0.9943	0.9956	0.9958	0.9959	0.9959	0.9960	0.9960	0.9956	0.9959	0.9961	0.9961	0.9960	0.9960	0.9961	0.9961	0.9962	0.9961	**0.9963**
	DM	13.7322	5.1982	5.6150	3.7245	4.0758	3.7105	2.6075	6.4839	4.3769	1.8974	0.2931	3.6074	3.6091	3.2288	3.4136	0.4904	-0.1199	-
	Sig. Level	***	***	***	***	***	***	***	***	***	*	None	***	***	***	***	None	None	-

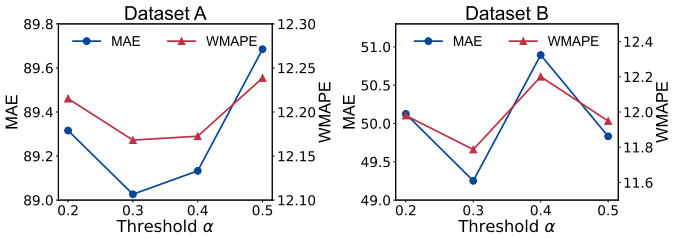


Fig. 5. Sensitivity analysis results for SCC thresholds α.

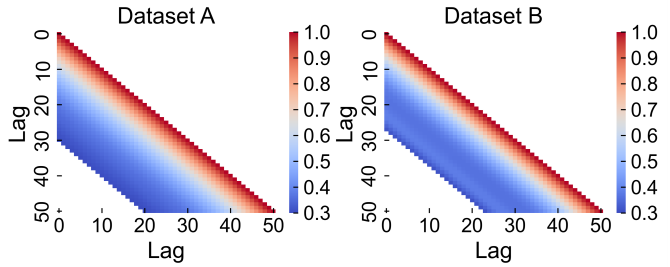


Fig. 6. Plots of SCC values on Dataset A and Dataset B.

Dataset A and Dataset B are shown in **Fig. 6**. It is observed that as the lag order increases, the correlation becomes smaller. This occurs because the correlation between the target data and the historical data decreases as the time interval increases. The appropriate input lengths for Datasets A, B, C, and D are 30, 27, 32, and 28, respectively.

E. Case I: Wind power forecasting with MG-DS

After determining the optimal input length, the proposed model MG-DS and all benchmark models were trained using training samples. The forecasting results of different models on Dataset A, B, C, and D are shown in **Table II**.

As illustrated in **Table II**, MG-DS achieved the best forecasting performance in terms of all metrics among all forecasting models. For Dataset C, MG-DS achieved the lowest MAE of 26.8003, RMSE of 37.3362, and WMAPE of 3.6829, along with the highest R^2 of 0.9927 and IA of 0.9982. Wind power forecasts of different models are shown in **Fig. 7**. Further experimental findings are summarized as follows.

(1) AI-based models outperformed the two statistical models on all datasets. Across all datasets, the persistence model exhibited the poorest performance across all metrics, underscoring the effectiveness of the other benchmark models in wind power forecasting. This suggests that patterns or regularities exist within the wind power series, which can be learned for effective forecasting. In comparison to AR, AI-based models generally exhibited performance enhancements. This may be attributed to AI-based models' more advanced feature extraction capabilities, which enable them to better capture the complex fluctuations inherent in wind power data.

(2) Popular ensemble strategies did not consistently enhance the wind power forecasting performance. A comparison between MG and its ensemble variants revealed that MG-V only achieved improvements in five metrics on Dataset C, while MG-Ba showed similar outcomes. In contrast, MG-Bo presented advancements on both Dataset A and B. Furthermore, MG-S exhibited improvements in MAE, RMSE, WMAPE, and R^2 on Dataset A and C, with quantifiable enhancements of 0.3681%, 0.5430%, 0.3690%, and 0.0412% for Dataset A, and 1.1143%, 0.3969%, 1.1162%, and 0.0101% for Dataset C, respectively. These findings emphasize that the effectiveness of ensemble strategies can significantly vary depending on the specific characteristics and complexities of the data.

(3) The MLP-based nonlinear feature extraction enabled the comprehensive mining of information from model inputs, leading to a more accurate description of the complex fluctuations in wind power. In comparison to GRU, MG's introduction of an MLP-based nonlinear feature extraction led to superior performance in all metrics on all datasets. On Dataset A, MG's improvements over GRU were 0.9390% in MAE, 0.5071% in MSE, 0.0309% in R^2, and 0.0101% in IA. Similarly, on Dataset D, MG outperformed GRU by 2.7290% in MAE, 1.8383% in MSE, 0.0508% in R^2, and 0.0100% in IA. While

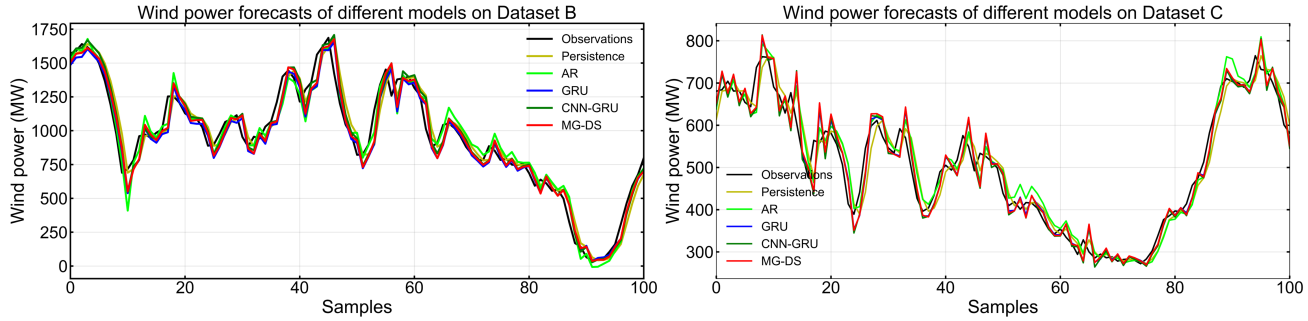


Fig. 7. Wind power forecasts of different models.

TABLE III
WIND POWER FORECASTING RESULTS OF THE DSSE PLUGIN FOR RNN-TYPE AND NON-RNN-TYPE BASE MODELS.

Type	Metrics	Dataset A			Dataset B			Dataset C			Dataset D		
		LSTM	GRU	LG-DSSE	LSTM	GRU	LG-DSSE	LSTM	GRU	LG-DSSE	LSTM	GRU	LG-DSSE
RNN	MAE	101.6225	101.0491	99.4705	52.07	51.6024	50.3812	27.9465	27.4633	27.3089	73.7632	74.1467	72.7152
	RMSE	145.0784	145.4119	144.4836	72.747	70.9291	70.4704	38.8288	38.1677	37.9706	117.4935	116.9615	114.6266
	WMAPE	8.8316	8.7818	8.6446	9.5291	9.4435	9.22	3.8404	3.774	3.7528	5.5645	5.5934	5.4854
	R^2	0.9707	0.9706	0.9709	0.9725	0.9738	0.9742	0.9921	0.9924	0.9925	0.9839	0.9841	0.9847
	IA	0.9925	0.9925	0.9927	0.9929	0.9932	0.9934	0.998	0.9981	0.9981	0.996	0.996	0.9961
Type	Metrics	Dataset A			Dataset B			Dataset C			Dataset D		
		$BPNN_c$	CNN_c	$BC\text{-}DSSE_c$	$BPNN_c$	CNN_c	$BC\text{-}DSSE_c$	$BPNN_c$	CNN_c	$BC\text{-}DSSE_c$	$BPNN_c$	CNN_c	$BC\text{-}DSSE_c$
Non-RNN	MAE	104.022	103.8232	103.0858	53.6884	52.6486	51.6803	29.2864	28.6734	28.2305	78.1655	76.8231	76.5011
	RMSE	149.7735	149.4546	147.9395	74.7149	72.671	72.1041	41.4305	39.7617	39.9729	120.8458	119.4068	117.5771
	WMAPE	9.0401	9.0229	8.9588	9.8253	9.635	9.4578	4.0246	3.9403	3.8795	5.8965	5.7953	5.771
	R^2	0.9688	0.9689	0.9695	0.971	0.9725	0.9729	0.991	0.9917	0.9916	0.983	0.9834	0.9839
	IA	0.992	0.9921	0.9922	0.9926	0.993	0.9932	0.9978	0.9979	0.9979	0.9957	0.9958	0.9959

MG performed relatively better compared to the model CNN-GRU on most datasets, it showed relative weaknesses in MAE and WMAPE on Dataset C.

(4) The proposed DS-based end-to-end ensemble method effectively enhanced forecasting performance and outperformed popular ensemble methods. The incorporation of the DS-based self-ensemble method in MG-DS resulted in substantial performance improvement compared to MG across the four datasets. Notably, MG-DS obtained the most significant enhancements on Dataset C, with improvements of 2.1194% in MAE, 1.3298% in MSE, 2.1208% in WMAPE, 0.0202% in R^2, and 0.0100% in IA, respectively. Among the five ensemble variants of MG (i.e., MG-V, MG-Ba, MG-Bo, MG-S, MG-DS), MG-DS exhibited the best forecasting performance.

The modified Diebold-Mariano (DM) test [48], based on the MAE metric, is used to assess whether the forecasting errors of the proposed model differ significantly from those of the benchmark models, providing a measure of their relative performance. Table II presents the DM statistics and their associated significance levels across four datasets. The significance levels are denoted by ***, **, *, and None, representing significance at the 1%, 5%, 10%, and above 10% thresholds, respectively. These levels are determined using p-values calculated from the DM statistics. The results of the DM test reveal that the proposed model achieves statistically significant improvements in forecasting accuracy over the benchmark models in most cases. This suggests that the proposed model provides more reliable and accurate forecasts, offering a meaningful advancement in predictive performance across the evaluated datasets.

F. Case II: Wind power forecasting with the DSSE plugin

The MG-DS's Module III can also be used as a DSSE plugin, allowing it to be applied to ensemble trained models, including both RNN-type and non-RNN-type models.

1) The DSSE plugin for trained RNN-type models

The DSSE plugin is applied to trained RNN models in a manner similar to the proposed model. Specifically, two trained RNN models (LSTM and GRU) in Case I were merged to form an ensemble model LG-DSSE with the proposed plugin. During this process, the parameters of the trained LSTM and GRU remained frozen, while only the plugin's parameters were further trained and optimized. The experimental results in **Table III** show that the DSSE plugin significantly improved the forecasting performance of the trained RNN-type models, demonstrating its adaptability and effectiveness across different model architectures. These findings reinforce the versatility of the DS-based self-ensemble strategy, highlighting its potential applications and generalizability across various model types.

2) The DSSE plugin for trained non-RNN-type models

The DSSE plugin can also be used to the trained non-RNN-type models. The BPNN and CNN models with the same hyperparameters in Case I were retrained with modified output forms as BPNNc and CNNc, respectively, and then

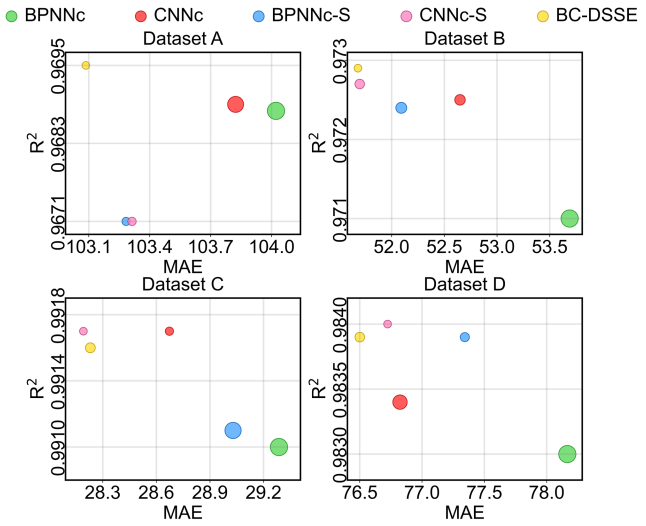


Fig. 8. Results of the DSSE plugin on non-RNN-type base models. (The smaller the point, the smaller the RMSE.)

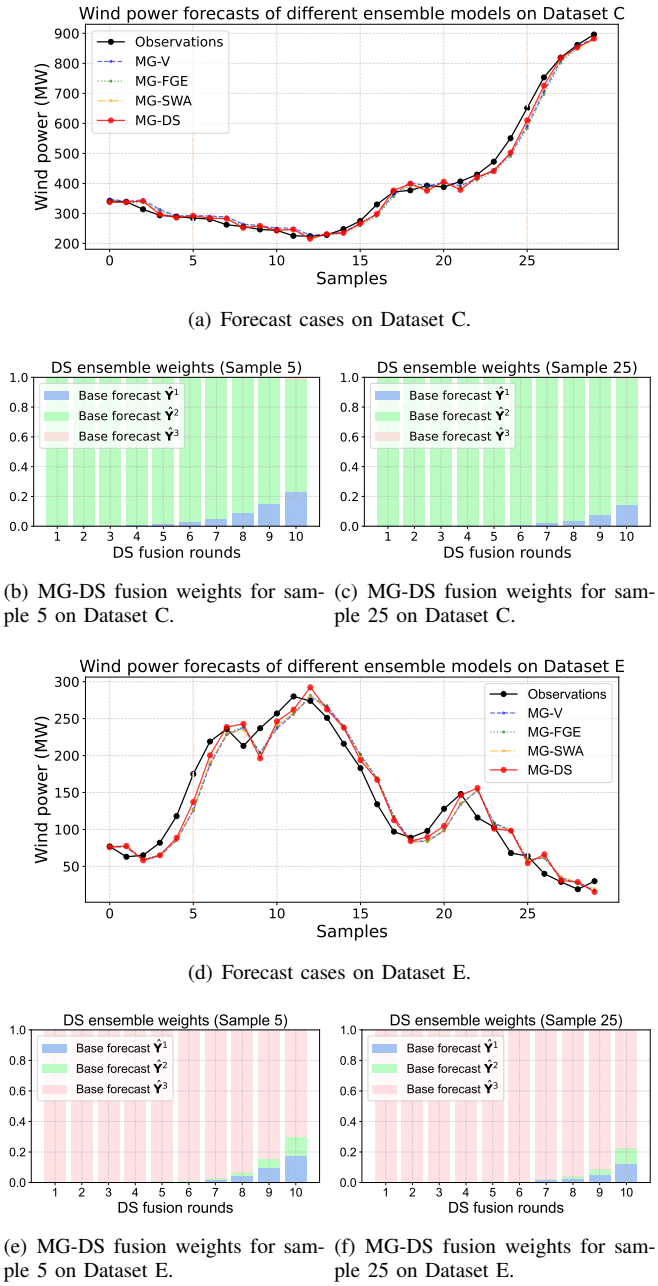
used as the base forecasters to construct an ensemble model BC-DSSE. The experimental results in **Table III** show that CNNc outperformed BPNNc, and BC-DSSE further improved the forecasting accuracy, and achieving the highest IA of 0.9979 on Dataset C. As shown in **Fig. 8**, BC-DSSE was also compared with BPNNc-S and CNNc-S, which ensemble BPNNs and CNNs using the snapshot ensemble strategy, respectively. The results show that BC-DSSE exhibited better or comparable performance compared to BPNNc-S and CNNc-S. These findings fully verify that the DS-based self-ensemble strategy is also applicable to the modified non-RNN-type forecasters and can improve the base forecasting performance.

VII. Discussion

A. Visualization of forecast cases and DS-based fusion weights

Fig. 9 compares forecast results and illustrates the corresponding dynamics of DS-based evidence fusion weights. As shown in **Fig. 9** (a) and (d), MG-DS generates predictions that align more closely with the ground truth than those of MG-V, MG-FGE, and MG-SWA. **Fig. 9** (b), (c), (e), and (f) illustrate the fusion weights of the base forecasts for two samples from Dataset C and E, respectively. The three base forecasts correspond to the outputs ($\hat{\mathbf{Y}}^1$, $\hat{\mathbf{Y}}^2$, and $\hat{\mathbf{Y}}^3$) generated by Module II in **Fig. 2**. There are two key findings: (1) The weight allocation varies across samples and datasets. For instance, Dataset C predominantly emphasizes $\hat{\mathbf{Y}}^2$, whereas Dataset E assigns greater weight to the $\hat{\mathbf{Y}}^3$. Moreover, even within the same dataset, different time points yield distinct weight distributions, reflecting the data-dependent adaptability of MG-DS. (2) The fusion weights evolve dynamically over the rounds. In Dataset C, increased fusion depth leads to a growing influence of base forecast $\hat{\mathbf{Y}}^1$ and the eventual emergence of a minor contribution from $\hat{\mathbf{Y}}^3$. In Dataset E, the fusion process progressively integrates $\hat{\mathbf{Y}}^1$ and $\hat{\mathbf{Y}}^2$.

These results demonstrate that MG-DS adaptively integrates base forecasts through multi-round DS-based fusion, resulting in improved prediction accuracy under varying conditions.



(a) Forecast cases on Dataset C.

(b) MG-DS fusion weights for sample 5 on Dataset C.

(c) MG-DS fusion weights for sample 25 on Dataset C.

(d) Forecast cases on Dataset E.

(e) MG-DS fusion weights for sample 5 on Dataset E.

(f) MG-DS fusion weights for sample 25 on Dataset E.

Fig. 9. The visualization of forecast cases and the corresponding ensemble weighting dynamics.

B. Effectiveness of DS-based magnifying glass in Module III

MG-CA was designed to evaluate the impact of DS-based magnifying glass by removing this layer from MG-DS. MG-MU represents MG-DS with the DS-based magnifying glass replaced by a mixup-based data augmentation method [49]. The experimental results in **Fig. 10** demonstrate that (1) the addition of the ensemble learning module enhances the forecasting performance of MG-CA compared to MG across four wind power datasets, (2) the removal of the DS-based layer leads to a significant decline in the forecasting performance of MG-CA compared to MG-DS, and (3) the mixup-based data augmentation method performs worse than MG-DS in forecasting accuracy. This verifies that the DS-based layer

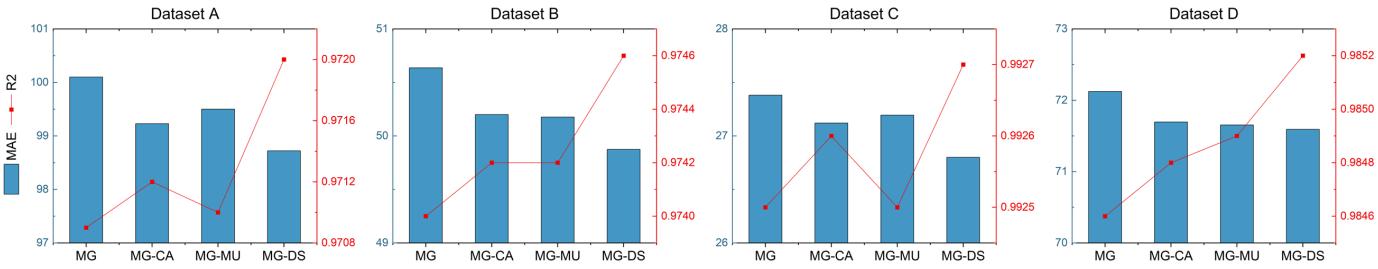


Fig. 10. Ablation experiment results for the DS-based magnifying glass in Module III on four datasets.

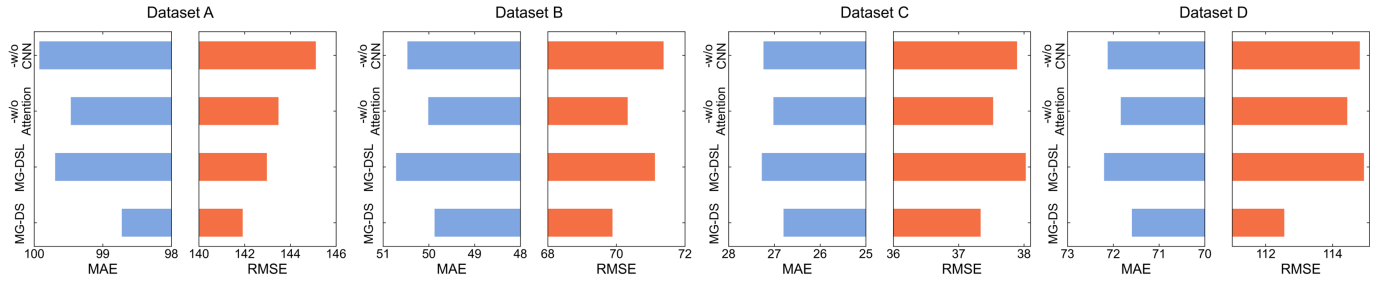


Fig. 11. Ablation experiment results for CNN and MHSA-based forecaster in Module III on four datasets.

effectively enhances the diversity of base forecasts in ensemble learning, thereby improving overall forecasting performance.

C. Effectiveness of CNN and MHSA in Module III

The impact of CNN and MHSA mechanism in the proposed DS-based self-ensemble forecasting module (Module III) were studies in this subsection. MG-DS was tested with the MHSA mechanism removed (-w/o Attention) and the CNN component excluded (-w/o CNN) separately. MG-DSL represents the CNN and MHSA-based forecaster is replaced with the Linear layer. The results, as shown in **Fig. 11**, indicate a consistent pattern across all datasets. From **Fig. 11**, the removal of the attention mechanism resulted in a slight decrease in model performance. This suggests that the attention mechanism positively contributes to the forecasting accuracy of MG-DS. The removal of the CNN component resulted in a more pronounced decrease in forecasting performance, indicating that the CNN component plays a significant role in the overall effectiveness in Module III of MG-DS. The replacement of both the CNN and MHSA mechanisms in MG-DSL with a linear layer caused a significant decline in the forecasting performance of MG-DS, highlighting the effectiveness of the CNN and MHSA mechanisms in Module III of MG-DS. Overall, the ablation study clearly demonstrates that both CNN and MHSA mechanism are crucial for achieving optimal results, with CNN playing a more dominant role in enhancing model accuracy and reliability.

D. Forecasting performance of MG-DS for a small wind plant

The forecasting performance of various models for the small wind plant (Dataset E) is presented in **Table IV**. In contrast to the other datasets, the maximum wind power of Dataset E is below 1000 MW. The statistical characteristics of Dataset E are provided in **Table I**. The results indicate

TABLE IV
COMPARISON OF FORECASTING RESULTS ON DATASET E.

Type	Model	MAE	RMSE	WMAPE	R^2	IA
Statistical models	Persistence	22.4348	188.9232	11.9075	0.9410	0.9851
	AR	23.1179	33.2207	12.2700	0.9381	0.9843
Machine learning models	RF	22.2664	32.2211	11.8181	0.9418	0.9844
	BPNN	21.7982	31.4570	11.5696	0.9445	0.9855
Deep learning models	CNN	21.8014	31.4059	11.5713	0.9447	0.9856
	LSTM	21.7983	31.5383	11.5696	0.9442	0.9856
	GRU	21.5459	31.1212	11.4357	0.9457	0.9859
	Mamba	21.2968	30.8962	11.3035	0.9465	0.9864
	DLinear	21.0073	30.1451	11.1498	0.9490	0.9866
Hybrid models	CNN-GRU	20.9297	30.3591	11.1086	0.9483	0.9866
	MG	20.8226	30.2731	11.0518	0.9486	0.9869
Ensemble models	MG-V	20.8145	30.2327	11.0475	0.9487	0.9868
	MG-Ba	20.7259	30.1455	11.0005	0.9490	0.9868
	MG-Bo	21.1198	30.3175	11.2095	0.9485	0.9864
	MG-S	20.7043	30.0640	10.9890	0.9493	0.9869
	MG-FGE	20.6350	30.0486	10.9522	0.9494	0.9870
	MG-SWA	20.6260	30.0933	10.9474	0.9492	0.9870
Ours	MG-DS	**20.2599**	**29.7747**	**10.7531**	**0.9503**	**0.9873**

that statistical models such as Persistence and AR perform less accurately compared to more advanced methods. Hybrid models, such as CNN-GRU and MG, demonstrate significant improvements by combining multiple approaches. Ensemble methods, including different variations of MG, further enhance forecasting accuracy and robustness by integrating several base forecasts. Overall, the proposed MG-DS model achieves the best forecasting performance across all metrics, outperforming other models. These results confirm that MG-DS is effective for forecasting wind power in low-power wind plants, emphasizing its robustness and generalizability across various wind power datasets.

VIII. CONCLUSION

To address the issue of increased training complexity and error accumulation in multi-stage ensemble methods for wind

power forecasting, this study proposed a simple but effective end-to-end ensemble model MG-DS. It combines base models learning and ensemble learning into a single process, leveraging all-MLP-based nonlinear feature extraction, GRU and CA-based base forecast generation, and DS-based self-ensemble forecasting. Moreover, a DSSE plugin was designed to ensemble already trained RNN-type or non-RNN-type base forecasters, providing an effective solution for adapting to existing forecasting models. The results on five wind power datasets demonstrated that our model generated more accurate wind power forecasts than traditional statistical models, AI-based models, and popular ensemble methods. Moreover, our proposed DS-based self-ensemble strategy, either embedded within the model or used as a standalone DSSE plugin, boosted the diversity of base forecasts, thereby improving the ensemble wind power forecasting performance.

Future work will focus on several key aspects. First, the proposed model could be improved by integrating more advanced techniques for feature extraction and evidence combination rules, offering potential for enhanced performance. Second, the current deterministic forecasting approach could be extended to support probabilistic forecasting, enabling the model to address a broader range of forecasting scenarios. Third, expanding the hyperparameter search space and utilizing continuous Bayesian optimization to explore a continuous parameter space could refine the tuning process, leading to improved model performance and optimization.

REFERENCES

[1] S. Boadu and E. Otoo, "A comprehensive review on wind energy in africa: Challenges, benefits and recommendations," *Renewable and Sustainable Energy Reviews*, vol. 191, p. 114035, 2024.

[2] L. Yang, Y. Chen, and X. Ma, "A state-age-dependent opportunistic intelligent maintenance framework for wind turbines under dynamic wind conditions," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 10, pp. 10 434–10 443, 2023.

[3] M. Yang, C. Ju, Y. Huang, Y. Guo, and M. Jia, "Short-term power forecasting of wind farm cluster based on global information adaptive perceptual graph convolution network," *IEEE Transactions on Sustainable Energy*, vol. 15, no. 3, pp. 2063–2076, 2024.

[4] Z. Zhang, J. Wang, D. Wei, T. Luo, and Y. Xia, "A novel ensemble system for short-term wind speed forecasting based on two-stage attention-based recurrent neural network," *Renewable Energy*, vol. 204, pp. 11–23, 2023.

[5] Y. Tang, K. Yang, S. Zhang, and Z. Zhang, "Wind power forecasting: A hybrid forecasting model and multi-task learning-based framework," *Energy*, vol. 278, p. 127864, 2023.

[6] E. Ahn and J. Hur, "A short-term forecasting of wind power outputs using the enhanced wavelet transform and arimax techniques," *Renewable Energy*, vol. 212, pp. 394–402, 2023.

[7] M. Li, M. Yang, Y. Yu, and W.-J. Lee, "A wind speed correction method based on modified hidden markov model for enhancing wind power forecast," *IEEE Transactions on Industry Applications*, vol. 58, no. 1, pp. 656–666, 2021.

[8] K. Wang, Y. Zhang, F. Lin, J. Wang, and M. Zhu, "Nonparametric probabilistic forecasting for wind power generation using quadratic spline quantile function and autoregressive recurrent neural network," *IEEE Transactions on Sustainable Energy*, vol. 13, no. 4, pp. 1930–1943, 2022.

[9] Z. Meng, Y. Guo, and H. Sun, "An adaptive approach for probabilistic wind power forecasting based on meta-learning," *IEEE Transactions on Sustainable Energy*, vol. 15, no. 3, pp. 1814–1833, 2024.

[10] L. Li, Z. Cen, M. Tseng, Q. Shen, and M. H. Ali, "Improving short-term wind power prediction using hybrid improved cuckoo search arithmetic-support vector regression machine," *Journal of Cleaner Production*, vol. 279, p. 123739, 2021.

[11] Y. Liu and J. Wang, "Transfer learning based multi-layer extreme learning machine for probabilistic wind power forecasting," *Applied Energy*, vol. 312, p. 118729, 2022.

[12] M. Li, M. Yang, Y. Yu, P. Li, and Q. Wu, "Short-term wind power forecast based on continuous conditional random field," *IEEE Transactions on Power Systems*, vol. 39, no. 1, pp. 2185–2197, 2024.

[13] J. Zhang, C. Cheng, and S. Yu, "Recognizing the mapping relationship between wind power output and meteorological information at a province level by coupling gis and cnn technologies," *Applied Energy*, vol. 360, p. 122791, 2024.

[14] J. Wang, H. Zhu, Y. Zhang, F. Cheng, and C. Zhou, "A novel prediction model for wind power based on improved long short-term memory neural network," *Energy*, vol. 265, p. 126283, 2023.

[15] M. Gong, C. Yan, W. Xu, Z. Zhao, W. Li, Y. Liu, and S. Li, "Short-term wind power forecasting model based on temporal convolutional network and informer," *Energy*, vol. 283, p. 129171, 2023.

[16] L. V. Krannichfeldt, Y. Wang, T. Zufferey, and G. Hug, "Online ensemble approach for probabilistic wind power forecasting," *IEEE Transactions on Sustainable Energy*, vol. 13, no. 2, pp. 1221–1233, 2022.

[17] I. Karijadi, S.-Y. Chou, and A. Dewabharata, "Wind power forecasting based on hybrid ceemdan-ewt deep learning method," *Renewable Energy*, vol. 218, p. 119357, 2023.

[18] 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*, vol. 15, no. 1, pp. 263–275, 2024.

[19] C. Chen and H. Liu, "Dynamic ensemble wind speed prediction model based on hybrid deep reinforcement learning," *Advanced Engineering Informatics*, vol. 48, p. 101290, 2021.

[20] H. Liu, C. Chen, X. Lv, X. Wu, and M. Liu, "Deterministic wind energy forecasting: A review of intelligent predictors and auxiliary methods," *Energy Conversion and Management*, vol. 195, pp. 328–345, 2019.

[21] J. Shi, C. Li, and X. Yan, "Artificial intelligence for load forecasting: A stacking learning approach based on ensemble diversity regularization," *Energy*, vol. 262, p. 125295, 2023.

[22] D. Kim and J.-G. Baek, "Bagging ensemble-based novel data generation method for univariate time series forecasting," *Expert Systems with Applications*, vol. 203, p. 117366, 2022.

[23] Y. Yang, H. Lv, and N. Chen, "A survey on ensemble learning under the era of deep learning," *Artificial Intelligence Review*, vol. 56, no. 6, pp. 5545–5589, 2023.

[24] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan, "Ensemble deep learning: A review," *Engineering Applications of Artificial Intelligence*, vol. 115, p. 105151, 2022.

[25] J. Shi, C. Li, and X. Yan, "Artificial intelligence for load forecasting: A stacking learning approach based on ensemble diversity regularization," *Energy*, vol. 262, p. 125295, 2023.

[26] H. Wang, Z. Tan, Y. Liang, F. Li, Z. Zhang, and L. Ju, "A novel multi-layer stacking ensemble wind power prediction model under tensorflow deep learning framework considering feature enhancement and data hierarchy processing," *Energy*, vol. 286, p. 129409, 2024.

[27] Y. Dong, H. Zhang, C. Wang, and X. Zhou, "Wind power forecasting based on stacking ensemble model, decomposition and intelligent optimization algorithm," *Neurocomputing*, vol. 462, pp. 169–184, 2021.

[28] G. Cirac, J. Farfan, G. D. Avansi, D. J. Schiozer, and A. Rocha, "Deep hierarchical distillation proxy-oil modeling for heterogeneous carbonate reservoirs," *Engineering Applications of Artificial Intelligence*, vol. 126, p. 107076, 2023.

[29] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21–45, 2006.

[30] H. Wu and D. Levinson, "The ensemble approach to forecasting: A review and synthesis," *Transportation Research Part C: Emerging Technologies*, vol. 132, p. 103357, 2021.

[31] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, pp. 1189–1232, 2001.

[32] Z. Wang, R. Wang, J. Gao, Z. Gao, and Y. Liang, "Fault recognition using an ensemble classifier based on Dempster-Shafer theory," *Pattern Recognition*, vol. 99, p. 107079, 2020.

[33] M. Tabassian, R. Ghaderi, and R. Ebrahimpour, "Combining complementary information sources in the Dempster-Shafer framework for solving classification problems with imperfect labels," *Knowledge-Based Systems*, vol. 27, pp. 92–102, 2012.

[34] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, "Snapshot ensembles: Train 1, get M for free," in *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France, 2017.

[35] T. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson, "Loss surfaces, mode connectivity, and fast ensembling of dnns," in *Proceedings of 32nd Conference on Advances in Neural Information Processing Systems*, vol. 31, Montréal, Canada, 2018, pp. 8803–8812.
[36] P. Izmailov, D. Podoprikhin, T. Garipov, D. P. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," in *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence*, Monterey, California, USA, 2018, pp. 876–885.
[37] A. P. Dempster, "Upper and lower probabilities induced by a multivalued mapping," in *Classic Works of the Dempster-Shafer Theory of Belief Functions*. Springer, 2008, pp. 57–72.
[38] G. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
[39] A. Laouafi, F. Laouafi, and T. E. Boukelia, "An adaptive hybrid ensemble with pattern similarity analysis and error correction for short-term load forecasting," *Applied Energy*, vol. 322, p. 119525, 2022.
[40] D. Yang, J.-e. Guo, S. Sun, J. Han, and S. Wang, "An interval decomposition-ensemble approach with data-characteristic-driven reconstruction for short-term load forecasting," *Applied Energy*, vol. 306, p. 117992, 2022.
[41] T. Abe, E. K. Buchanan, G. Pleiss, R. Zemel, and J. P. Cunningham, "Deep ensembles work, but are they necessary?" in *Proceedings of the 36th Conference on Advances in Neural Information Processing Systems*, vol. 35, New Orleans, USA, 2022, pp. 33 646–33 660.
[42] H. Guo, Y. Zhang, and W. Wang, "Dynamical targeted ensemble learning for streaming data with concept drift," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 12, pp. 8023–8036, 2024.
[43] A. Das, W. Kong, A. Leach, S. K. Mathur, R. Sen, and R. Yu, "Long-term forecasting with tiDE: Time-series dense encoder," *Transactions on Machine Learning Research*, 2023.
[44] A. Bampoulas, F. Pallonetto, E. Mangina, and D. P. Finn, "A bayesian deep-learning framework for assessing the energy flexibility of residential buildings with multicomponent energy systems," *Applied Energy*, vol. 348, p. 121576, 2023.
[45] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.
[46] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, Washington, DC, USA, 2023, pp. 11 121–11 128.
[47] Z. Zhao, S. Yun, L. Jia, J. Guo, Y. Meng, N. He, X. Li, J. Shi, and L. Yang, "Hybrid vmd-cnn-gru-based model for short-term forecasting of wind power considering spatio-temporal features," *Engineering Applications of Artificial Intelligence*, vol. 121, p. 105982, 2023.
[48] D. Harvey, S. Leybourne, and P. Newbold, "Testing the equality of prediction mean squared errors," *International Journal of Forecasting*, vol. 13, no. 2, pp. 281–291, 1997.
[49] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, BC, Canada, 2018.