Aalborg Universitet

Ultra-Short-Term Interval Prediction of Wind Power Based on Graph Neural Network and Improved Bootstrap Technique

Liao, Wenlong; Wang, Shouxiang; Bak-Jensen, Birgitte; Pillai, Jayakrishnan Radhakrishna; Yang, Zhe; Liu, Kuangpu

Ultra-short-term Interval Prediction of Wind Power Based on Graph Neural Network and Improved Bootstrap Technique

Wenlong Liao, Shouxiang Wang, Birgitte Bak-Jensen, Jayakrishnan Radhakrishna Pillai, Zhe Yang, and Kuangpu Liu

Abstract—Reliable and accurate ultra-short-term prediction of wind power is vital for the operation and optimization of power systems. However, the volatility and intermittence of wind power pose uncertainties to traditional point prediction, resulting in an increased risk of power system operation. To represent the uncertainty of wind power, this paper proposes a new method for ultra-short-term interval prediction of wind power based on a graph neural network (GNN) and an improved Bootstrap technique. Specifically, adjacent wind farms and local meteorological factors are modeled as the new form of a graph from the graph-theoretic perspective. Then, the graph convolutional network (GCN) and bi-directional long short-term memory (Bi-LSTM) are proposed to capture spatiotemporal features between nodes in the graph. To obtain high-quality prediction intervals (PIs), an improved Bootstrap technique is designed to increase coverage percentage and narrow PIs effectively. Numerical simulations demonstrate that the proposed method can capture the spatiotemporal correlations from the graph, and the prediction results outperform popular baselines on two real-world datasets, which implies a high potential for practical applications in power systems.

Index Terms—Wind power, graph neural network (GNN), bi-directional long short-term memory (Bi-LSTM), prediction interval, Bootstrap technique.

I. INTRODUCTION

NORMALLY, the ultra-short-term prediction of wind power refers to the estimation of wind power with the time horizon, which ranges from a few minutes to several hours [1]. Ultra-short-term prediction of wind power has a significant impact on the safe and economic operation (e.g., real-time dispatch planning) of power systems because of the risks associated with their fluctuation and intermittence

[2]. Therefore, there is a need to develop accurate ultra-short-term prediction methods of wind power [3].

Generally, ultra-short-term prediction of wind power consists of two components: deterministic point prediction and error estimation. The works of deterministic point prediction fall under three headings: physical methods, statistical methods, and artificial intelligence (AI) based methods.

1) Physical methods rely on the information of surrounding wind field (e.g., obstacle, surface roughness, and terrain) and numerical weather prediction (NWP) data (e.g., humidity, pressure, wind speed, and temperature) to model the relationship between wind power and wind speeds [4]. The physical methods are suitable for ultra-short-term prediction of new wind farms or wind turbines, since historical data are not required to train the model. However, the detailed physical parameters and NWP data bring a severe computational burden [5]. In addition, wrong meteorological parameter is easy to accumulate errors of the physical methods, which seriously affects the prediction accuracy.

2) Statistical methods mainly include auto-regressive (AR), auto-regressive integrated moving average (ARIMA), auto-regressive moving average (ARMA), and gray methods [6], which employ historical wind power to predict future wind power. Although the calculation speeds of these methods are very fast, most of them have limited prediction accuracy, especially for the wind power generation curves with strong stochastic nature (e. g., prominent peaks and steep ramps). This is because these methods ignore the correlation between wind power and meteorological factors [7].

3) Support vector machine (SVM), light gradient boosting machine (LightGBM), and multi-layer perceptron (MLP) are widely-used AI-based methods for ultra-short-term prediction of wind power in the last 20 years [8]. Compared with physical methods, SVM, LightGBM, and MLP are more cost-effective, but they have difficulty in capturing the temporal correlation of wind power generation curves accurately. To solve this problem, a variety of deep neural networks (DNNs) have been proposed recently. In particular, recurrent neural networks (RNNs) [9], such as long short-term memory (LSTM) and gated recurrent unit (GRU), have shown outstanding performance in modeling the temporal dependence of wind power generation curves, which significantly improves the ultra-short-term prediction accuracy of wind power.

The traditional point prediction aims to generate deterministic prediction values, which cannot represent the prediction error caused by various reason such as volatility and intermittence of wind power. Further, interval prediction is one of the mainstream ways to estimate the error by adding lower and upper boundaries to each deterministic prediction value. The popular methods to construct prediction intervals (PIs) mainly include the Delta [10], Bayesian [11], Gaussian [12], mean-variance estimation [13], lower upper bound estimation (LUBE) [14], and Bootstrap technique [15]. Specifically, the first four methods [10]-[13] are restricted with specified probability distributions of prediction errors assumed artificially, since numerous factors (e.g., input data, point prediction model, and time horizon of prediction) affect the probability distribution of prediction error, which is difficult to be formulated accurately in most cases. The LUBE employs a DNN with two outputs to calculate the lower and upper boundaries of PIs, but the design of loss functions suitable for the gradient descent method remains a challenge when training the DNN [16]. The Bootstrap technique is a flexible and efficient way, which iteratively resamples historical prediction errors to generate PIs without any distribution assumptions of prediction errors. So far, the Bootstrap technique has been widely used for interval prediction of renewable energy sources and loads because of its simple process and outstanding performance [17]. All the same, a great limitation of the traditional Bootstrap technique remains to be solved. For each prediction value, the Bootstrap technique has a very close PI width. In theory, if the number of resamples is infinite, the PI width of each prediction is the same. Ideally, the perfect PIs should be narrow when the prediction error is small, and PIs should be wide when the prediction error is large. In other words, the similar widths of PIs constructed by the traditional Bootstrap technique have difficulty in balancing the coverage percentage and width of PIs.

In a broad sense, the inputs of wind power prediction should be considered as a graph [18]. Specifically, geographically adjacent wind farms and local meteorological factors are represented as nodes of the graph whose adjacency matrix can represent the spatial correlation between nodes. Historical data are the features of nodes, which can model the temporal correlation of time series. However, the traditional point prediction models (e.g., MLP, LightGBM, and LSTM) defined in the Euclidean domain cannot deal with the graph, so few publications have been oriented from a graph perspective in the past few years. Usually, traditional point prediction models have to simplify the graph into Euclidean data by ignoring the adjacency matrix, which adversely affects the prediction accuracy. This simplification makes it difficult for the traditional point prediction models defined in the Euclidean domain to capture the spatial correlation between multiple adjacent wind farms, limiting the prediction accuracy.

There has been increasing interests in generalizing traditional DNNs into graph neural networks (GNNs) in recent years. In particular, graph convolutional networks (GCNs) have been widely used in different fields (e.g., link prediction, drug synthesis, and traffic flow prediction) due to their superiority in modeling the spatial correlation between nodes [19]. Although GNNs have great potential for wind power prediction whose inputs are regarded as a graph, the applications of GNNs for wind power prediction are relatively limited. In [20], GCNs are applied to model the relationships between offshore wind farms. To capture spatial and temporal correlations between multiple wind nodes, a GCN and an LSTM are integrated in [21] and [22]. However, these previous publications [18], [20]-[22] only model adjacent wind farms as nodes and ignore meteorological factors. In other words, they are not suitable for ultra-short-term prediction considering meteorological factors for individual wind farm. Besides, they do not involve the uncertainty of wind power. Especially, GNNs have rarely been applied to interval prediction such as ultra-short-term interval prediction of wind power.

Based on the above discussion, this paper proposes a novel GNN-based point prediction model and an improved Bootstrap technique for ultra-short-term interval prediction of wind power. Specifically, a GCN is employed to model the spatial correlation between nodes, and a more recent advanced model named bi-directional long short-term memory (Bi-LSTM) is utilized to capture the temporal correlation of time-series curves. Then, an improved Bootstrap technique is designed to balance the coverage percentage and width of PIs. Finally, the effectiveness of the proposed method is verified through real datasets. The main difference between this paper and previous publications involving GNNs lies in:

1) The nodes are generalized from adjacent wind farms into both wind farms and meteorological factors.

2) Different from previous publications [21], [22] without considering the uncertainty of wind power, this paper extends the GNN from point prediction into the interval prediction to account for the uncertainty.

3) The performance of point prediction model is improved by applying bidirectional learning techniques into the traditional LSTM, i.e., Bi-LSTM replaces the traditional LSTM to capture temporal correlations.

The key contributions of this paper are summarized as follows.

1) Without simplifying the inputs of ultra-short-term prediction of wind power into Euclidean data, this paper innovatively attempts to model the inputs as the new form of a graph from a graph-theoretic perspective. The spatial correlation between nodes is represented by an adjacency matrix. The historical data are viewed as the features of nodes to describe the temporal correlation of the wind power generation curves and meteorological factors.

2) To improve the accuracy of the point prediction, a novel GNN combining the GCN and Bi-LSTM is proposed to capture spatiotemporal correlations without artificial feature engineering.

3) As a flexible and efficient way, the improved Bootstrap technique is proposed to balance the coverage percentage and width of PIs. Besides, it is free of any distribution assumptions of prediction errors.

4) Extensive numerical simulations on two real-world data-

sets are performed to validate the effectiveness of the proposed method for ultra-short-term interval prediction of wind power.

The rest of this paper is organized as follows. Section II proposes a novel GNN for wind power prediction. Section III presents the improved Bootstrap technique and introduces the commonly-used evaluation indices of PIs. Section IV tests the proposed method and popular baselines on real datasets. Section V discusses the proposed method. Finally, the conclusion is given in Section VI.

II. A NOVEL GNN FOR WIND POWER PREDICTION

Normally, interval prediction includes two steps: deterministic point prediction and error estimation. In this section, the predictive information (i.e., wind power of multiple wind farms and nearby meteorological factors) is represented as an undirected graph. Then, a GCN and a Bi-LSTM are integrated to model spatiotemporal correlations for point predictions, whose prediction errors are represented in the next section through the improved Bootstrap technique.

A. Problem Definition

Normally, ultra-short-term prediction of wind power is performed using wind power of multiple wind farms and surrounding meteorological factors as inputs to a point prediction model. In other words, each wind farm is represented by its wind power, rather than the physical model.

As one of the innovations, this subsection employs a simple undirected graph $G=(V,E)$ to represent multiple wind farms and surrounding meteorological factors [23] (e. g., wind speed, temperature, and humidity), as shown in Fig. 1. Note that meteorological factors are generally collected from the supervisory control and data acquisition (SCADA) system or surrounding weather stations. Specifically, the wind power of the wind farm is viewed as a real node, and each meteorological factor is considered as a virtual node. All nodes of this undirected graph can be represented as $V=\{v_1,v_2,\ldots,v_n\}$, where v_i is the ith node; and n is the total number of real nodes and virtual nodes. The features at time t of the graph can be expressed as $X_t^g=\{X_t^{v_1},X_t^{v_2},\ldots,X_t^{v_n}\}$, where $X_t^{v_i}$ is the feature of the ith node at time t.

•Wind power of wind farm (real node); ── Correlation coefficient (edge)
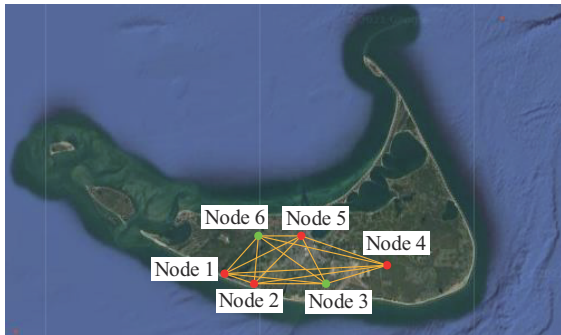 • Meteorological factor (virtual node)

Fig. 1.   Simple undirected graph to represent multiple wind farms and surrounding meteorological factors.

In practice, the predictive information is not always available. For example, some datasets without meteorological factors only include real nodes, and some datasets with one wind farm and surrounding meteorological factors only include a real node and multiple virtual nodes.

With respect to social networks, the correlation between nodes is generally described by an adjacency matrix A consisting of 0 and 1, where 0 means there is not an edge; and 1 means there is an edge. Similarly, the adjacency matrix of graph for wind farms and surrounding meteorological factors can be emulated with a correlation matrix $C \in \mathbf{R}^{n \times n}$ to model the spatial dependence between nodes. There may exist different ways to construct graphs, which may be explored in future works due to page limits. For example, multiple wind farms can be constructed as a directed graph if the dataset includes only wind power without meteorological factors. However, the inputs of wind power prediction normally include wind power and meteorological factors. The wind power of the wind farm is viewed as a real node, and each meteorological factor is considered as a virtual node. It is difficult to describe the direction between real nodes and virtual nodes. Therefore, the undirected graph is constructed to describe the correlation (i.e., edge) between nodes.

As a simple example, the widely-used Pearson correlation coefficient $C(v_i,v_j,t)$ is employed to represent the distance (i.e., edge) between the ith node and the jth node at time t as:

$$C\left(v_i,v_j,t\right) = \frac{\left|\sum_{l=0}^{h}\left(X_{t-l}^{v_i}-\bar{X}^{v_i}\right)\left(X_{t-l}^{v_j}-\bar{X}^{v_j}\right)\right|}{\sqrt{\sum_{l=0}^{h}\left(X_{t-l}^{v_i}-\bar{X}^{v_i}\right)^2}\sqrt{\sum_{l=0}^{h}\left(X_{t-l}^{v_j}-\bar{X}^{v_j}\right)^2}} \tag{1}$$

where $X_{t-l}^{v_i}$ and $X_{t-l}^{v_j}$ are the historical features of the ith and the jth nodes at time $t-l$, respectively; and $\bar{X}^{v_i}$ and $\bar{X}^{v_j}$ are the average features of the ith and the jth nodes from time $t-h$ to time t, respectively. Note that the correlation matrix is time-varying with features of the graph.

So far, the inputs of the point prediction model have been modeled as an undirected graph to capture the correlation between wind farms and surrounding meteorological factors.

Ultra-short-term point prediction aims to predict the wind power of the ith wind farm at time $t+k$ based on the historical features from time $t-h$ to time t and its correlation matrix $C(t)$. The outputs and inputs of GNNs can be expressed as (2) and (3), respectively.

$$\hat{X}_{t+k}^{v_i}= GNN\left(C\left(t\right),X_{\text{feature}}\right) \tag{2}$$

$$X_{\text{feature}}= \left(X_t^g,X_{t-1}^g,\ldots,X_{t-h}^g\right) \tag{3}$$

where k is the time horizon; $\hat{X}_{t+k}^{v_i}$ is the predicted wind power of the ith wind farm at time $t+k$; and $X_{\text{feature}} \in \mathbf{R}^{n \times h}$ is the feature matrix of the graph from time $t-h$ to time t, and each node has h features. Note that (2) represents one-step prediction, which can be generalized into multi-step prediction by modifying k and training multiple models.

In the next subsections, a novel GNN is proposed to model the spatiotemporal correlations of wind farms and meteorological factors, as shown in Fig. 2. Firstly, the correlation

matrix and feature matrix of nodes are used as inputs of GCN layers to represent topological information of the graph for modeling the spatial features [24]. Then, the time series with spatial features obtained from the GCN are fed into Bi-LSTM layers, which capture temporal features through information transmission among input gate, a forget gate, an output gate, and a cell state. In the end, two dense layers are employed to output $X_{t+k}^{v_i}$.
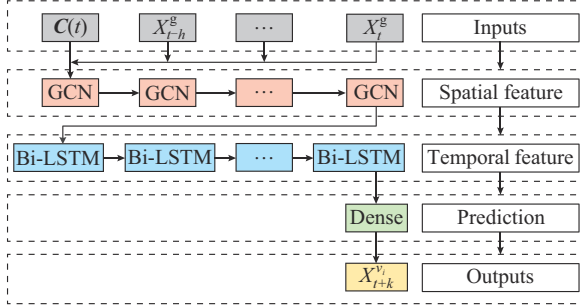

Fig. 2. Framework of proposed GNN.

B. GCN

It is a vital issue to model the complex spatial dependencies between nodes for ultra-short-term prediction of wind power. Traditional CNN can only extract local spatial features of data (e.g., images) defined in the Euclidean domain, while the input data of ultra-short-term prediction of wind power are a graph rather than 2-dimensional matrices, which means that traditional CNN cannot capture complex topological information and spatial dependencies between nodes of the graph. Fortunately, the traditional CNN has been extended into the GCN defined in the graph domain to handle graph-structured data, and has received more and more attention because of its powerful performances.

There are many variants of GCN, which is mainly classified into two broad categories [25]: spectral-based GCN and spatial-based GCN. Among them, the spectral-based GCN maps the graph to a new space through the Fourier transform, and performs convolutional operations in the new space, just like the traditional CNN. Then, the data are mapped back to the graph domain to obtain spatial features. The calculation process of spatial-based GCN is relatively simple, since it directly defines convolutional operation based on the spatial correlation of nodes in the graph domain. In general, both spatial-based GCN and spectral-based GCN are developing and evolving rapidly, and it is difficult to identify which one performs better. Compared with the spatial-based GCN, the spectral-based GCN is more widely used because it was proposed earlier. Without loss of generality, the popular spectral-based GCN is employed to obtain the spatial features of inputs.

Given a correlation matrix $C(t)$ and a feature matrix $X_{feature}$, the graph convolutional layer captures the spatial features between nodes through its first-order polynomial in the Laplacian after constructing a filter in the Fourier domain. As shown in Fig. 3, a spectral-based GCN generally consists of multiple graph convolutional layers, which can be represented as:

$$\begin{cases} H_{GCN}^{(i)} = \sigma_g \left(\hat{C} H_{GCN}^{(i-1)} W_{GCN}^{(i-1)} \right) \\ H_{GCN}^{(0)} = X_{feature} \end{cases} \quad i = 1, 2, ..., n_g \quad (4)$$

$$\begin{cases} \hat{C} = D^{-\frac{1}{2}} \tilde{C} D^{-\frac{1}{2}} \\ \tilde{C} = C + I \\ D_{ii} = \sum_j \tilde{C}_{ij} \end{cases} \quad (5)$$

where I is the identity matrix; $\tilde{C}$ is a new form of correlation matrix with self-loop (the correlation matrix of each graph convolutional layer is the same); D is the degree matrix of the correlation matrix; n_g is the number of graph convolutional layers; $\sigma_g(\cdot)$ is the activation function of graph convolutional layers; $W_{GCN}^{(i)}$ represents the parameters to be optimized through supervised training of the ith graph convolutional layer; and $H_{GCN}^{(i)}$ represents the outputs. Note that the time series with spatial features obtained from the GCN are considered as inputs of the Bi-LSTM in the following subsection.
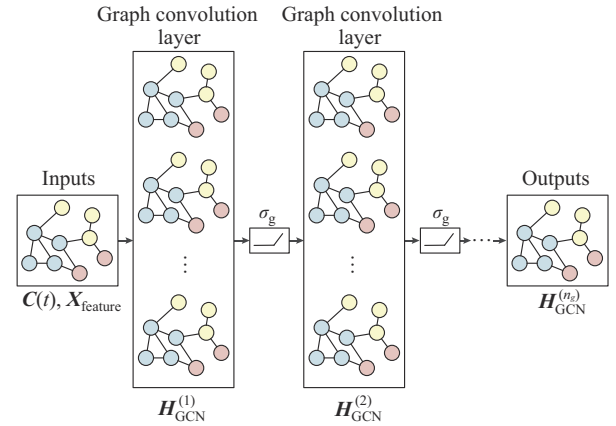
Fig. 3. Framework of spectral-based GCN consisting of multiple graph convolutional layers.

C. Bi-LSTM

Another key issue to ultra-short-term prediction of wind power is modeling temporal dependence. Traditional DNNs (e.g., MLP) are incompatible for modeling time-series data, while RNN is a very promising algorithm, which is proficient in processing time-series data such as audio signals. Considering the traditional RNN involves vanishing gradient problems, some excellent variants have been proposed and show outstanding performance in different fields [26]. Therefore, a recent advanced variant (e.g., Bi-LSTM layers) is employed to capture temporal features of time series from the last GCN layer.

Figure 4 shows the structure of a simple LSTM unit, which consists of an input gate, a forget gate, an output gate, and a cell state. The cell state memorizes the values over different time lengths, and the above-mentioned three gates adjust the data flow into and out of the cell state. The relationship between input and output of LSTM is as follows.
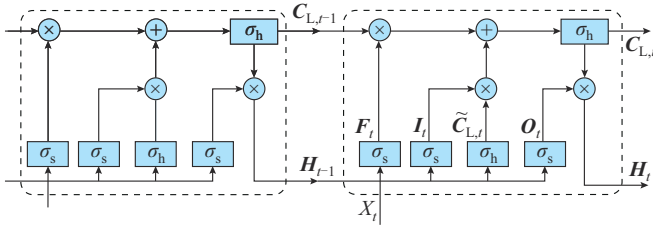
Fig. 4.   Structure of LSTM unit.

$$
\begin{cases}
\tilde{C}_{\mathrm{L},t} = \sigma_{\mathrm{h}}\left(W_{\mathrm{C}} \cdot \left[H_{\mathrm{L},t}, H_{t-1}\right] + B_{\mathrm{C}}\right) \\
F_t = \sigma_{\mathrm{s}}\left(W_{\mathrm{F}} \cdot \left[H_{\mathrm{L},t}, H_{t-1}\right] + B_{\mathrm{F}}\right) \\
O_t = \sigma_{\mathrm{s}}\left(W_{\mathrm{O}} \cdot \left[H_{\mathrm{L},t}, H_{t-1}\right] + B_{\mathrm{O}}\right) \\
I_t = \sigma_{\mathrm{s}}\left(W_{\mathrm{I}} \cdot \left[H_{\mathrm{L},t}, H_{t-1}\right] + B_{\mathrm{I}}\right) \\
C_{\mathrm{L},t} = F_t \odot C_{\mathrm{L},t-1} + I_t \odot \tilde{C}_{\mathrm{L},t} \\
H_t = O_t \odot \sigma_{\mathrm{h}}\left(C_{\mathrm{L},t}\right)
\end{cases}
\tag{6}
$$

where O_t, I_t, F_t, and $\tilde{C}_{\mathrm{L},t}$ are the activation vectors of the output gate, input gate, forget gate, and cell input activation, respectively; $\sigma_{\mathrm{s}}(\cdot)$ and $\sigma_{\mathrm{h}}(\cdot)$ are the sigmoid function and hyperbolic tangent function, respectively; W_{O} is the weight of the output gate; W_{F} is the weight of the forget gate; W_{C} is the weight of the cell state; W_{I} is the weight of the update gate; B_{F}, B_{I}, B_{O}, and B_{C} are the bias vectors of the forget gate, input gate, output gate, and cell state, respectively; H_t is the latent state vector at time t; $X_{\mathrm{L},t}$ is the feature information at time t; $C_{\mathrm{L},t}$ is the cell state vector at time t; and $\odot$ is the Hadamard product.

Bidirectional learning is a widely-used technique to improve the prediction accuracy of traditional LSTM for sequence learning tasks, since the output of time-series prediction is not the only product of the previous input data, but a continuously correlated component. Bidirectional learning can help LSTM capture the temporal features in bidirectional aspects (i.e., the forward and reverse paths), while traditional LSTM is trained to model temporal features in one-way data flow (i.e., the forward path) only. The LSTM with bidirectional learning technique shows higher performance than traditional LSTM in various sequence learning tasks such as audio signal processing. Therefore, the Bi-LSTM is presented to capture the temporal features of time series from the last GCN layer.

As shown in Fig. 5, the forward LSTM is used to model the relationship among feature information at time t, latent state vector at time $t-1$, and the cell state vector at time $t-1$, while the backward LSTM is employed to combine the feature information at time t, latent state vector at time $t+1$, and cell state vector at time $t+1$. The mathematical equations of the Bi-LSTM are expressed as:

$$
\begin{cases}
\vec{H}_t = LSTM\left(H_{t-1}, C_{\mathrm{L},t-1}, X_{\mathrm{L},t}\right) \\
\overleftarrow{H}_t = LSTM\left(H_{t+1}, C_{\mathrm{L},t+1}, X_{\mathrm{L},t}\right) \\
H_{\mathrm{Bi},t} = \sigma_{\mathrm{Bi}}\left(\vec{H}_t, \overleftarrow{H}_t\right)
\end{cases}
\tag{7}
$$

where $\vec{H}_t$ is the latent state vector at time t of the forward LSTM; $\overleftarrow{H}_t$ is the latent state vector at time t of the backward LSTM; $H_{\mathrm{Bi},t}$ is the latent state vector at time t of the Bi-LSTM; and $\sigma_{\mathrm{Bi}}(\cdot)$ is a mathematical operator (e.g., summation, multiplication, and concatenation) that is used to combine $\overleftarrow{H}_t$ and $\vec{H}_t$.
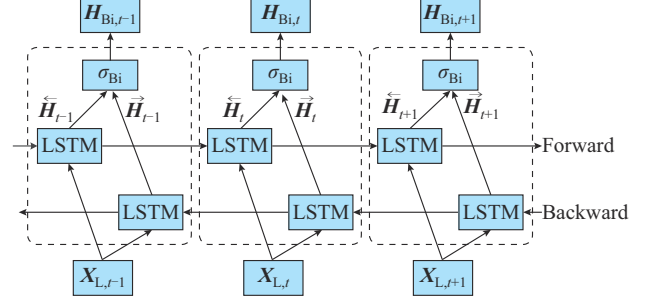
Fig. 5.   Structure of Bi-LSTM unit.

D. Dense Layer

Finally, the temporal features obtained from multiple Bi-LSTM layers are used as the inputs of a dense layer at the end of the GNN, which outputs the predicted wind power of the ith wind farm at time $t+k$:

$$
\hat{X}_{t+k}^{v_i} = \sigma_{\mathrm{d}}\left(X_{\mathrm{d}} W_{\mathrm{d}} + B_{\mathrm{d}}\right)
\tag{8}
$$

where X_{d} is the vector of inputs of the dense layer; W_{d} and B_{d} are the vectors of weights and biases of the dense layer, respectively; and $\sigma_{\mathrm{d}}(\cdot)$ is the activation function of the dense layer.

III. CONSTRUCTION OF PIs

A deterministic point prediction model is proposed in the previous section. In this section, the traditional Bootstrap technique is improved to represent the prediction errors using narrow PIs. Then, several evaluation indices of PIs are presented.

A. Traditional Bootstrap Technique

A traditional deterministic point prediction only provides a single point that hides the error of wind power from noises of the dataset and the model itself, while interval prediction is an effective way to quantify the uncertainty through a lower and upper boundary. The PIs surround the prediction value from the deterministic point prediction model and cover the real value with high probability.

Bootstrap is a robust technique for error estimation, which can be used to generate PIs without making any assumption about the functional form of the probability distribution of prediction errors. Specifically, the construction of PIs for ultra-short-term prediction of wind power using the traditional Bootstrap technique mainly includes two steps [27].

1) Error Estimation of Training Set

A pre-trained point prediction model and real wind power are used to obtain the prediction errors of the training set. Then, the prediction errors of the training set are employed to construct the PIs of point predictions for the test set. In

particular, the prediction errors of the training set are randomly sampled n_t times into the group 1, where n_t is the number of Bootstrap repeats, which should be large enough to ensure meaningful statistics. Ideally, the Bootstrap repeats are often hundreds or thousands given the time resources. In this paper, n_t is 5000. A prediction error is obtained for each sampling process. Note that the Bootstrap technique allows a prediction error of the training set to be sampled more than once (i.e., sampling with replacement).

2) Construction of PIs

The prediction errors in group 1 are sorted in descending order, and the values at the given percentile α are considered as the PI nominal confidence (PINC). For instance, when α is equal to 0.9, a confidence interval of 90% PINC can be obtained by selecting the errors at the 95% percentile as the upper boundary and the 5% percentile as the lower boundary.

B. Improved Bootstrap Technique

Although Bootstrap is a widely powerful and applicable statistical technique for quantifying uncertainty, its PIs are too conservative. Figure 6 presents a simple example of wind power interval prediction with a prediction time horizon of 1 hour, which includes a real wind power generation curve, point predictions generated from the GNN, and PIs constructed from traditional Bootstrap technique.
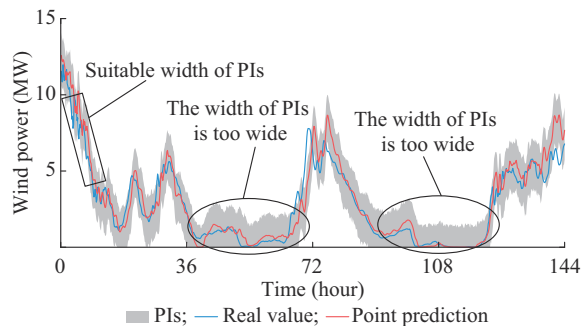

Fig. 6.   A simple example of wind power interval prediction.

For each wind power, the traditional Bootstrap technique constructs PIs with a fixed interval width. The wide PIs are suitable for the periods when the wind power generation curve is highly volatile, as shown in the area surrounded by a rectangular. However, these fixed PIs are obviously too wide for wind power with small volatility (e.g., the area enclosed by ellipses), which will lead to the lack of concentration of PIs. Too wide PIs are also called conservative PIs, that is when wide PIs are used for risk-based decision-making (e.g., interval optimization) of power systems, their solutions require more reserve capacity of generation sides, giving a negative impact on economics. In short, the fixed PIs of the traditional Bootstrap technique is a great limitation, which remains to be solved.

Ideally, appropriate PIs should be narrow when the wind power is weakly volatile, as weak volatility tends to imply small prediction errors. Relatively, PIs should be wide, when wind power is highly volatile, because highly volatile often means large prediction errors. To evaluate the volatility of predicted wind power at time $t+k$, the widely-used standard

deviation is employed as:

$$S_{t+k}^{v_i} = Std\left(\hat{X}_{t+k}^{v_i}, \hat{X}_{t+k-1}^{v_i}, ..., \hat{X}_{t+k-q}^{v_i}\right) \qquad (9)$$

where $S_{t+k}^{v_i}$ is the standard deviation of predicted wind power of the ith wind farm from time $t+k-q$ to time $t+k$. Note that the size of q is determined to be 7 in this paper by analyzing the Pearson correlation coefficient between standard deviation and prediction errors. For other datasets, q may vary, but it can also be determined by analyzing the Pearson correlation coefficient.

In the previous paragraph, the standard deviations of point predictions are employed to reflect the volatility and prediction errors of wind power. In other words, if the standard deviation is large, the prediction error is also large. If the standard deviation is small, the prediction error is also small.

To valid the relationship between the prediction errors and the standard deviation of point predictions, the proposed GNN is used to obtain point predictions for the test set. Then, the standard deviation of each point prediction is calculated by (9), and the prediction error of each point prediction can be obtained by calculating the absolute error. Finally, the standard deviation of each point prediction is considered as the *x*-axis, and the prediction error of each point prediction is regarded as the *y*-axis, as shown in Fig. 7.
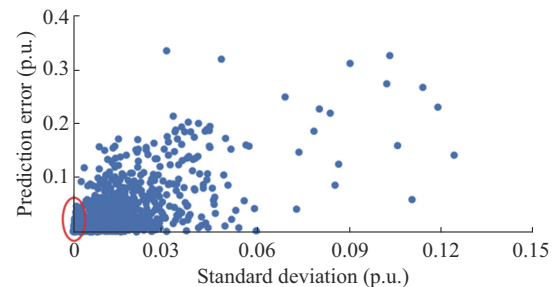
Fig. 7.   Prediction error and standard deviation.

From Fig. 7, the following conclusions can be found.

1) When the standard deviation is small (i.e., the volatility is weak), the point prediction errors are also small. In other words, a small standard deviation implies a small prediction error. For example, for the region surrounded by ellipses in Fig. 7, the standard deviation of point prediction is tiny, corresponding to a very small prediction error.

2) As the standard deviation becomes larger, some large prediction errors start to appear, and the large prediction errors are distributed at the locations where the standard deviation is greater than 0.03 p.u.. At these points, PIs should be wide to cover real values.

Generally, point prediction errors are strongly correlated with standard deviations, which should have the potential to guide the design of PIs. In brief, when the standard deviations of point predictions are small, the PIs should be narrow, because the prediction errors are small. When the standard deviations of point predictions are large, the PIs should be wide, since the prediction error may be large.

Based on the above analysis, this paper improves the traditional Bootstrap technique using standard deviation to obtain appropriate PIs. Firstly, the prediction errors are grouped

based on the standard deviation of point predictions. For point predictions in the test set, standard deviations are calculated to determine which group they belong to. Then, this paper resamples the prediction errors from the same grouped validation set for point predictions of the test set.

A vivid explanation is that the improved Bootstrap technique narrows the PIs of the ellipse-enclosed regions in Fig. 6 without changing the PIs of the other areas. The specific implementation steps of the improved Bootstrap technique are as follows.

1) Grouping Prediction Errors and Standard Deviation of Validation Set

Normally, compared with the training set, prediction errors of the validation set are closer to those of the test set. Therefore, the prediction errors of validation set are used to estimate the prediction errors of test set in the improved Bootstrap technique. For each point prediction of the validation set, its prediction error and standard deviation are calculated. This prediction error is put into group 1. Meanwhile, if the standard deviation is lower than s_1, this prediction error is also placed in group 2. Obviously, group 2 is the proper subset of group 1.

2) Group Assignment

For each point prediction of the test set, its standard deviation is calculated using (9). If this standard deviation is less than s_2, assign all prediction errors in group 2 to group 3; otherwise, assign all prediction errors in group 1 to group 3. Then, the prediction errors in group 3 are randomly sampled n_t times, and the sampled prediction errors are put into group 4. Note that the prediction error in group 3 can be sampled more than once (i.e., sampling with replacement).

Theoretically, s_1 should be greater than s_2. If s_1 is smaller than s_2, prediction errors of group 1 are smaller than the real point prediction errors for the test set, so the constructed PIs will be too narrow to cover real values.

For example, if s_1 is 0.01, all prediction errors in group 1 are less than 0.15, which can be observed from Fig. 7 (i.e., when the standard deviation is less than 0.01, the maximum value of the prediction error is smaller than 0.15). Further, if s_2 is 0.06 and the standard deviation of a point prediction is 0.04, the prediction errors in group 1 will be used to construct the PIs according the second step of the improved Bootstrap technique. Obviously, this is not reasonable. The reason is that the standard deviation of the point prediction is 0.04, and its real prediction error may be greater than 0.3, as shown in Fig. 7. However, the maximum error in group 1 is less than 0.15, so the constructed PIs cannot cover the real value.

3) Construction of PIs

The prediction errors in group 4 are sorted in descending order, and the values at the given percentile α are considered as the confidence interval. Specifically, the error at the $(100\alpha + 100((1-\alpha)/2))^{\text{th}}$ percentile is the upper boundary, and the $(50(1-\alpha))^{\text{th}}$ percentile is the lower boundary.

For ease of understanding and reproducing the improved Bootstrap technique, the Algorithm 1 in Appendix A shows the codes and comments directly with MATLAB.

C. Evaluation Indices of PIs

The evaluation of PIs is often considered in terms of reliability and sharpness [2], [3]. Specifically, reliability is measured by prediction interval coverage percentage (PICP) and sharpness is represented by prediction interval normalized average width (PINAW):

$$PICP = \frac{1}{N}\sum_{i=1}^{N} g_i \tag{10}$$

$$PINAW = \frac{1}{N}\sum_{i=1}^{N}(U_i - L_i) \tag{11}$$

where U_i is the upper boundary of the ith point prediction; and L_i is the lower boundary of the ith point prediction. If the real wind power is within PIs, $g_i = 1$; otherwise, $g_i = 0$.

Generally, the larger the PICP is, the more reliable the PIs are. With the same *PICP*, a smaller *PINAW* indicates narrower PIs and higher quality of PIs. In addition, *PICP* and *PINAW* are two conflicting metrics, so previous research often employs the coverage width criterion (CWC) to balance them [2], [3], [14], [28]:

$$CWC = PINAW\left(1 + \gamma\left(PICP, \alpha\right)e^{-\eta(PICP-\alpha)}\right) \tag{12}$$

$$\gamma(PICP, \alpha) = \begin{cases} 0 & PICP \geq \alpha \\ 1 & PICP < \alpha \end{cases} \tag{13}$$

where η is a penalty coefficient. The larger η is, the greater contributions of the PICP to the CWC are. Generally, η is determined by the system operator. As an example, η is equal to 5 in this paper according to [2], [3]. The smaller the CWC is, the better the performance of interval prediction is.

IV. CASE STUDY

A. Data Description and Simulation Conditions

To fully test the performance of the proposed GNN and improved Bootstrap technique, different features of two wind power datasets are used for simulation and discussion. Specifically, the first dataset includes the real wind power of 16 geographically adjacent wind farms without meteorological factors, so the first dataset can be considered as a graph with 16 real nodes whose features are the historical wind power. The second dataset consists of real wind power of a wind farm and local meteorological factors (e. g., wind direction, wind speed, temperature, pressure, and density), and the second dataset can also be viewed as a graph with 6 nodes (1 real node and 5 virtual nodes), whose features are represented by historical wind power or historical meteorological factors.

Wind power prediction is performed based on the following information: ① NWP information (i.e., forecasts of meteorological factors); ② historical meteorological factors and wind power. Their respective importance is highly dependent on the predicted time horizon. The shorter the time horizon is, the more impact historical measurements have, and vice versa for NWP information. Normally, the equilibrium is about 2 hours according to [29]. The NWP information for both datasets is not available due to various reasons, so this

paper focuses on the comparisons between the proposed method and baselines for ultra-short-term prediction of wind power within 2 hours. The NWP information can be easily included as nodes to validate ultra-short-term prediction of wind power for longer time horizon in future works.

Both datasets are collected by the National Renewable Energy Laboratory [23], [30], and they have a temporal resolution of 10 min for wind power and meteorological factors. The first dataset covers the period from March 2011 to February 2012, and the second one ranges from March 2008 to February 2009. Considering that variable weather conditions in different seasons contribute to the uncertainties of wind power, the proposed method is separately trained to test the performance in different seasons, including spring (March to May), summer (June to August), autumn (September to November), and winter (December to February). There is no optimal split percentage of the training set, validation set, and test set. The commonly used rough standard splits are 80%, 10%, and 10%, where 80% of the data are used to train the model for each season, the immediately following 10% of the data are utilized to determine the parameters of the model, and the final 10% of the data are employed to test the performance.

To validate the performance of the proposed method, some advanced deterministic point prediction models (e. g., MLP in [31], LightGBM in [8], GCN in [18], Bi-LSTM in [14], and GNN in [21]) are employed as baselines. The parameters of these models vary slightly for different regional and seasonal wind power datasets. In this paper, the model parameters for the different seasons are determined by the control variables in [32] - [34]. As a simple example, the main parameters of each model for the first dataset in spring are given as follows.

1) The middle layer of the MLP includes 3 dense layers, whose numbers of neurons are 30, 35, and 20, respectively.

2) For the LightGBM, the maximum tree depth is 5, and the number of the maximum tree leaves is 25; the number of boosted trees is 1000, and the learning rate is 0.001; the minimum number in a child is 80, and the subsample ratio is 0.8.

3) The middle layer of the GCN includes 3 GCN layers, whose numbers of output channels are 32, 16, and 16, respectively.

4) The middle layer of the Bi-LSTM includes 3 Bi-LSTM layers, whose dimensions of the output space are 25, 25, and 20, respectively.

5) As shown in Fig. 8, the middle layer of the proposed GNN includes 2 GCN layers and 2 Bi-LSTM layers, where the numbers of output channels of GCN layers are 32 and 16, respectively; and the dimensions of outputs of Bi-LSTM layers are 25 and 20, respectively.

6) For the GNN in [21], its structure is similar to the proposed model, but it replaces Bi-LSTM layers with traditional LSTM layers.

Besides, the following parameters are common to these models: the optimizer is the Adam algorithm, and the loss function is the mean absolute error. The training epoch is 200, and the batch size is 32. The activation function of each middle layer is rectified linear unit (ReLU) function. The output layer of each model is a dense layer with 1 neuron, and its activation function is the sigmoid function.
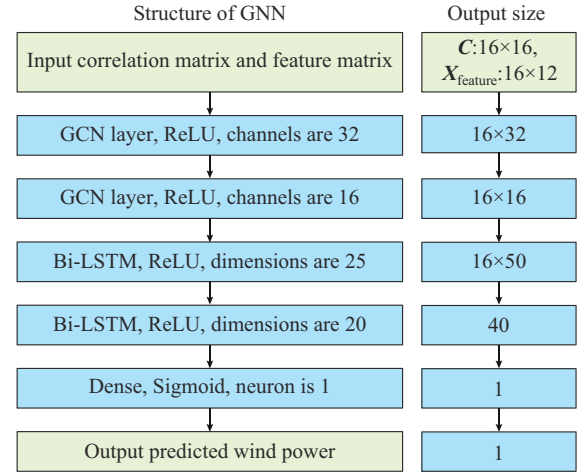
Fig. 8.   Parameters of proposed GNN.

All the above-mentioned models are tested in the Spyder 4.1.5 with the Spektral 1.0 and Tensorflow 2.0, which are popular libraries of deep learning. The key parameters of computer are as follows: CPU 1.60 GHz, 8 GB RAM, Intel Core$^{(TM)}$ i5-10210U.

B. Parameter Discussion of Improved Bootstrap Technique

To discuss the key parameters of the improved Bootstrap technique, two cases from the spring are used as simple examples to show how to select s_1 and s_2. This selection process provides general guidance. Repetition can reduce anomalous results. Considering limited time resources, the proposed GNN is independently trained 30 times. In addition, the difference of metrics between the traditional and improved Bootstrap techniques is presented to analyze whether key parameters have a positive or negative impact on performance.

Specifically, the traditional Bootstrap technique is used to obtain the average metrics of the validation set, recorded as $PICP_1$ and $PINAW_1$. Then, the improved Bootstrap technique with different parameters (s_1 and s_2 varying from 0.004 to 0.1) is employed to obtain the average metrics of the validation set, recorded as $PICP_2$ and $PINAW_2$. Finally, the differences between metrics $\Delta PICP = PICP_2 - PICP_1$ and $\Delta PINAW = PINAW_2 - PINAW_1$ are visualized, as shown in Fig. 9(a)-(d).

If $s_1 < s_2$, $PICP_2$ is also smaller than $PICP_1$, even though $PINAW_2$ is smaller than $PINAW_1$. In other words, narrow PIs come at the expense of PI accuracy. Relatively, when $s_1 > s_2$, $PICP_2$ is very close to $PICP_1$ and most of the $PINAW_2$ is smaller than $PINAW_1$ as we expect, since the improved Bootstrap technique outperforms the traditional one.

Further, the partial CWCs of the validation set are shown in Fig. 9(e) and (f), where $PICP_2$ is greater than $PICP_1$, and $PINAW_2$ is less than $PINAW_1$. For example, $s_1 = 0.036$ and $s_2 = 0.024$ are the suitable parameters for the first data set in spring. Generally, in order to reduce the width of PIs without reducing the $PICP$, s_1 should be larger than s_2, which is

consistent with the theoretical analysis in Section III-B. Conversely, the performance of improved Bootstrap technique may be inferior to that of traditional one, if $s_1 < s_2$. For other datasets, the sizes of s_1 and s_2 may vary (they can be explored by similar simulations), but s_1 must be greater than s_2.
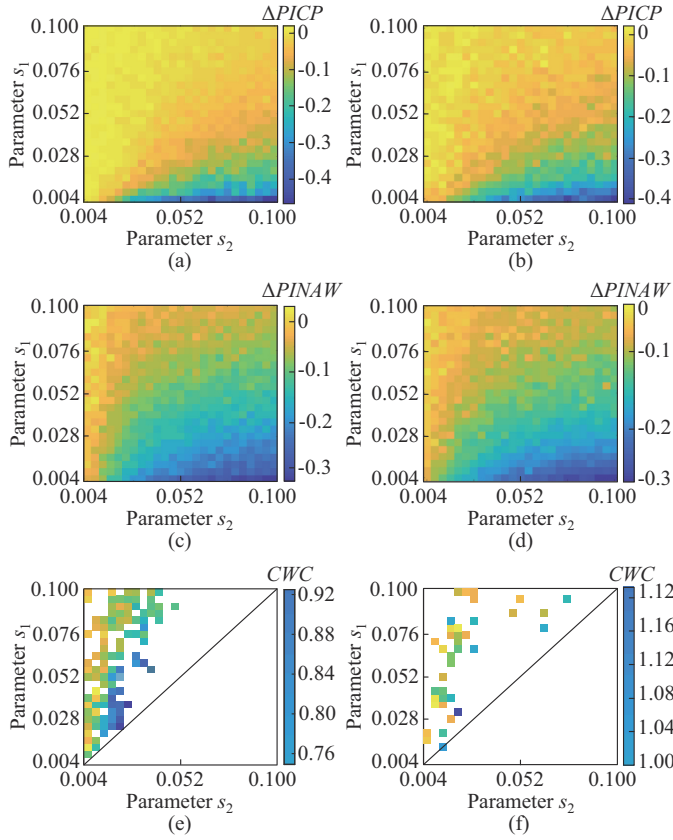

Fig. 9.   Average metrics of validation set. (a) $\Delta PICP$ of the first dataset. (b) $\Delta PICP$ of the second dataset. (c) $\Delta PINAW$ of the first dataset. (d) $\Delta PINAW$ of the second dataset. (e) CWC of the first dataset. (f) CWC of the second dataset.

C. Comparative Analysis of Interval Construction

The main goal of wind power interval prediction is to derive reliable and narrow PIs. Compared with low-confidence-level PIs, high-confidence-level PIs are more practically meaningful for economic and safe operation of power systems, so different confidence levels varying from 90% to 99% will be considered in following simulations.

To test the performance of the improved Bootstrap technique, the traditional Bootstrap technique [15] and Gaussian methods [12] are considered as baselines, which are used to construct PIs. Taking the 1-hour prediction time horizon as an example, the different average metrics of the test set in different seasons are given in Tables I-IV.

Although the PINAW of the Gaussian method is smaller than those of Bootstrap techniques, its PICP is much smaller than the PINC. The reason for this phenomenon is that the prediction errors of deterministic point prediction models do not follow the Gaussian distribution. From the above four tables, it is found that both traditional and improved Bootstrap techniques provide more reliable PIs of the measured wind power than the Gaussian method.

TABLE I
DIFFERENT AVERAGE METRICS OF TEST SET IN SPRING

Method	PINC (%)	The first dataset			The second dataset		
		PICP (%)	PINAW (p.u.)	CWC (p.u.)	PICP (%)	PINAW (p.u.)	CWC (p.u.)
Gaussian method	90	42.1	0.090	1.074	58.8	0.208	1.196
	95	49.2	0.107	1.164	68.7	0.248	1.176
	99	59.3	0.141	1.167	78.9	0.327	1.220
Traditional bootstrap	90	84.3	0.300	0.700	80.4	0.309	0.808
	95	91.1	0.424	0.939	87.1	0.404	1.004
	99	99.1	0.847	0.847	93.8	0.772	1.774
Improved bootstrap	90	84.4	0.274	0.637	81.1	0.283	0.725
	95	91.1	0.371	0.822	87.0	0.372	0.926
	99	98.7	0.627	1.262	96.1	0.633	1.364

TABLE II
DIFFERENT AVERAGE METRICS OF TEST SET IN SUMMER

Method	PINC (%)	The first dataset			The second dataset		
		PICP (%)	PINAW (p.u.)	CWC (p.u.)	PICP (%)	PINAW (p.u.)	CWC (p.u.)
Gaussian method	90	60.3	0.024	0.129	60.9	0.163	0.862
	95	64.2	0.028	0.161	68.6	0.195	0.929
	99	69.7	0.037	0.199	78.9	0.257	0.959
Traditional bootstrap	90	92.7	0.168	0.168	92.1	0.347	0.347
	95	97.0	0.248	0.248	96.8	0.466	0.466
	99	100.0	0.553	0.553	99.7	0.858	0.858
Improved bootstrap	90	92.3	0.109	0.109	91.9	0.323	0.323
	95	97.6	0.160	0.160	97.2	0.445	0.445
	99	99.7	0.314	0.314	99.2	0.716	0.716

TABLE III
DIFFERENT AVERAGE METRICS OF TEST SET IN AUTUMN

Method	PINC (%)	The first dataset			The second dataset		
		PICP (%)	PINAW (p.u.)	CWC (p.u.)	PICP (%)	PINAW (p.u.)	CWC (p.u.)
Gaussian method	90	57.8	0.127	0.761	51.7	0.173	1.348
	95	64.4	0.152	0.853	60.8	0.207	1.352
	99	73.7	0.200	0.907	76.6	0.272	1.104
Traditional bootstrap	90	85.7	0.215	0.482	82.5	0.314	0.771
	95	91.9	0.345	0.748	92.6	0.444	0.945
	99	98.3	0.637	1.296	98.1	0.741	1.518
Improved bootstrap	90	84.2	0.164	0.383	82.9	0.262	0.635
	95	92.6	0.268	0.571	93.0	0.391	0.823
	99	98.3	0.457	0.930	98.1	0.585	1.199

Note that the partial CWCs of the Gaussian method are smaller than those of Bootstrap techniques in some scenarios (e. g., the first dataset with 99% PINC in summer), but it does not mean that the Gaussian method is better than Bootstrap techniques. The reason is that the PICP of the Gaussian method is much smaller than the PINC. For example, in the first dataset with 99% PINC in summer, the PICP of the Gaussian method is 69.7%, whereas the expected probability to cover real values is 99%. Relatively, the PICPs of Boot-

strap techniques are greater than the PICNs in most scenarios, which indicates that Bootstrap techniques can ensure the security of the power system with the expected probability.

Method	PINC (%)	The first dataset			The second dataset		
		PICP (%)	PINAW (p.u.)	CWC (p.u.)	PICP (%)	PINAW (p.u.)	CWC (p.u.)
Gaussian method	90	47.8	0.100	0.921	59.5	0.115	0.646
	95	54.3	0.119	1.031	67.8	0.138	0.675
	99	63.8	0.157	1.069	77.9	0.184	0.712
Traditional Bootstrap	90	81.8	0.267	0.669	90.0	0.280	0.560
	95	91.5	0.368	0.806	94.9	0.425	0.852
	99	99.2	0.822	0.822	98.8	0.805	1.617
Improved Bootstrap	90	82.4	0.236	0.579	89.5	0.234	0.474
	95	91.5	0.316	0.692	94.2	0.331	0.676
	99	99.2	0.612	0.612	99.0	0.562	0.562

Further, the comparison of the PICP and PINAW between the traditional Bootstrap technique and the improved Bootstrap technique shows that their PICPs are very similar, but the PINAW of the latter is smaller, suggesting that the improved Bootstrap technique can effectively narrow the width of PIs with negligible reduction of PICP. For example, in the first dataset in winter, the PICP is 0.992 for both the traditional and improved Bootstrap techniques when the PINC is 99%, and the PINAW for the improved Bootstrap technique is reduced by 25.54% compared with the traditional Bootstrap technique.

In the first dataset with *PINC* = 99% in spring, the CWC of the improved Bootstrap technique is larger than that of the traditional Bootstrap technique. This is because the definition of CWC derived from [2], [3] is not reasonable in some extreme scenarios. The PICP of the improved Bootstrap technique is 98.7%, which is slightly less than 99%. The penalty coefficient in the definition makes the CWC increase sharply. In fact, compared with the traditional Bootstrap technique, the improved Bootstrap technique has only 0.4% lower PICP, while its PINAW is reduced by 25.97%. It is worthwhile to reduce the coverage percentage a little in exchange for very narrow PIs. In other words, the improved Bootstrap technique is better than the traditional Bootstrap technique for this case.

Figure 10 randomly selects two samples from the first and second datasets respectively to visually compare the traditional and improved Bootstrap techniques. Note that the prediction time horizon is still 1 hour.

It is clear that the traditional and improved Bootstrap techniques have similar PIs for the strong volatile regions (e.g., steep ramps, prominent peaks and valleys). This is because prediction errors in strong volatile regions tend to be large, and narrowing the PIs in these regions may cause the PICP to drop. The improved Bootstrap technique aims to keep the PIs in strong volatile regions and reduce the width of PIs in the weak volatile regions. For example, elliptical enclosed regions in Fig. 10 are less volatile and the PIs constructed by

the traditional Bootstrap method are too wide, whereas the improved Bootstrap technique effectively narrows the PIs in these regions. Based on the above comparisons, the improved Bootstrap technique with high PICPs and narrow PINAWs is applied for the proposed method and four baselines in next sub-sections.
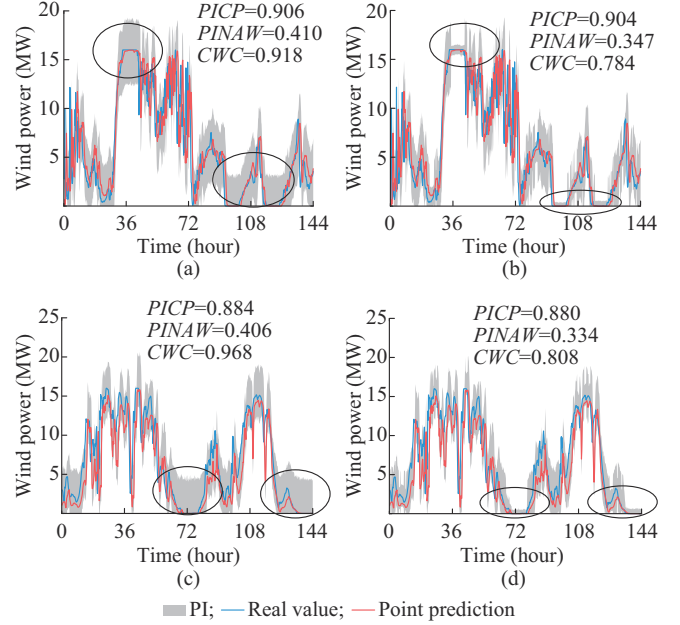
Fig. 10. Comparison of traditional and improved Bootstrap techniques using two samples from the first and second datasets. (a) Interval prediction using traditional Bootstrap technique in the first dataset. (b) Interval prediction using improved Bootstrap technique in the first dataset. (c) Interval prediction using traditional Bootstrap technique in the second dataset. (d) Interval prediction using improved Bootstrap technique in the second dataset.

D. Comparison Between Proposed Method and Baselines

In order to validate the superiority of the proposed method, the popular deterministic point prediction methods (e.g., MLP in [31], LightGBM in [8], GCN in [18], Bi-LSTM in [14], and GNN in [21]) are performed as baselines. The 1-hour prediction time horizon is used as an example. Each point prediction model is trained 30 times independently, and average metrics (e.g., PICP, PINAW, and CWC) of the test set are listed in Table V.

1) *Strong prediction performance:* it can be observed that the proposed method obtains the smallest CWCs under all PINCs and seasons for both two real datasets, proving the effectiveness for wind power interval prediction. For example, for the first dataset with 90% PINC in spring, the CWCs of the proposed method are 15.18%, 15.40%, 13.69%, 1.70%, and 1.85% lower than those of the MLP, LightGBM, GCN, Bi-LSTM, and GNN, respectively. Note that some methods may have higher PICPs compared with the proposed method, but they are at the cost of interval width, that is, their PINAWs are much larger than that of the proposed method. For example, for the first dataset with 99% PINC in summer, the PICPs and PINAWs of the Bi-LSTM are increased by 0.1% and 29.3% compared with the proposed method, respectively, which leads to smaller CWCs of the proposed method than the baselines.

TABLE V
AVERAGE METRICS OF DIFFERENT PREDICTION METHODS WITH DIFFERENT PINCs

Dataset	Season	PINC (%)	MLP			LightGBM			GCN			Bi-LSTM			GNN			Proposed method		
			PICP (%)	PINAW (p.u.)	CWC (p.u.)	PICP (%)	PINAW (p.u.)	CWC (p.u.)	PICP (%)	PINAW (p.u.)	CWC (p.u.)	PICP (%)	PINAW (p.u.)	CWC (p.u.)	PICP (%)	PINAW (p.u.)	CWC (p.u.)	PICP (%)	PINAW (p.u.)	CWC (p.u.)
First	Spring	90	80	0.282	0.751	80	0.286	0.753	84	0.316	0.738	86	0.290	0.648	85	0.285	0.649	84	0.274	0.637
		95	92	0.456	0.980	94	0.449	0.913	91	0.407	0.900	92	0.404	0.865	92	0.394	0.861	91	0.371	0.822
		99	99	0.780	1.574	98	0.769	1.561	99	0.726	1.456	98	0.668	1.386	98	0.645	1.313	99	0.627	1.262
	Summer	90	91	0.144	0.144	92	0.135	0.135	94	0.116	0.116	93	0.124	0.124	92	0.121	0.121	92	0.109	0.109
		95	93	0.434	0.903	99	0.270	0.270	98	0.205	0.205	97	0.191	0.191	97	0.175	0.175	98	0.160	0.160
		99	100	0.496	0.496	100	0.464	0.464	100	0.387	0.387	100	0.406	0.406	100	0.383	0.383	100	0.314	0.314
	Autumn	90	83	0.217	0.518	81	0.205	0.522	87	0.211	0.462	82	0.158	0.391	83	0.16	0.386	84	0.164	0.383
		95	87	0.368	0.921	89	0.340	0.827	91	0.330	0.740	92	0.376	0.803	92	0.333	0.716	93	0.268	0.571
		99	91	0.562	1.397	95	0.550	1.215	98	0.564	1.166	98	0.469	0.964	98	0.464	0.954	98	0.457	0.930
	Winter	90	85	0.312	0.712	81	0.262	0.681	83	0.284	0.695	84	0.285	0.672	83	0.271	0.656	82	0.236	0.579
		95	95	0.499	1.010	91	0.412	0.921	92	0.419	0.911	88	0.348	0.833	90	0.329	0.760	92	0.316	0.692
		99	98	0.688	1.405	99	0.710	0.710	99	0.695	0.695	99	0.697	0.697	99	0.672	0.672	99	0.612	0.612
Second	Spring	90	79	0.302	0.820	81	0.320	0.815	81	0.306	0.779	79	0.275	0.762	79	0.281	0.758	81	0.283	0.725
		95	88	0.448	1.097	86	0.415	1.083	89	0.447	1.048	88	0.413	1.005	87	0.394	0.973	87	0.372	0.926
		99	96	0.717	1.569	96	0.633	1.375	98	0.652	1.352	98	0.643	1.324	98	0.64	1.323	97	0.631	1.318
	Summer	90	87	0.329	0.714	95	0.396	0.396	93	0.372	0.372	92	0.363	0.363	92	0.354	0.354	92	0.323	0.323
		95	94	0.427	0.878	98	0.509	0.509	98	0.518	0.518	97	0.485	0.485	97	0.477	0.477	97	0.445	0.445
		99	100	0.784	0.784	100	0.764	0.764	99	0.741	0.741	100	0.748	0.748	100	0.742	0.742	99	0.716	0.716
	Autumn	90	83	0.329	0.805	83	0.287	0.692	84	0.295	0.692	80	0.248	0.666	80	0.248	0.653	83	0.262	0.635
		95	92	0.414	0.887	90	0.388	0.893	94	0.428	0.877	92	0.388	0.850	92	0.39	0.841	93	0.391	0.823
		99	99	0.714	1.440	98	0.644	1.328	98	0.645	1.312	97	0.605	1.290	97	0.612	1.285	98	0.585	1.199
	Winter	90	87	0.255	0.560	87	0.251	0.550	88	0.236	0.502	83	0.200	0.489	84	0.223	0.483	90	0.234	0.474
		95	95	0.408	0.825	91	0.339	0.761	94	0.370	0.751	92	0.333	0.711	93	0.332	0.699	94	0.331	0.676
		99	99	0.637	1.277	99	0.596	1.207	98	0.461	0.951	99	0.586	0.586	99	0.579	0.579	99	0.562	0.562

2) Spatiotemporal prediction capability: compared with the traditional MLP and LightGBM, neural network-based models (e. g., GCN, Bi-LSTM, GNN, and proposed GNN), which aim to model temporal features or spatial features, usually have better precision for ultra-short-term interval prediction of wind power. For instance, for the first dataset with 90% PINC in spring, the PICPs of the GCN, Bi-LSTM, GNN, and the proposed method are increased by approximately 4.4%, 6.0%, 5.3%, and 4.6% compared with the MLP, and the CWCs are 1.73%, 13.72%, 13.58%, and 15.18% lower than that of the MLP, respectively. The CWCs of the GCN, Bi-LSTM, GNN in [21], and the proposed method are approximately reduced by 1.99%, 13.94%, 13.81%, and 15.41% than that of the LightGBM, and their PICPs are improved by 4.0%, 5.6%, 4.9%, and 4.2%, respectively. This is mainly because MLP and LightGBM have difficulties in dealing with complex and non-stationary wind power data and meteorological factors. Besides, the performances of GCN and Bi-LSTM are limited and unable to be further improved, since GCN can only consider spatial features and ignore temporal features, while Bi-LSTM can only account for temporal features and neglect spatial features. Note that the GNN in [21] consists of both GCN and LSTM, while the proposed method replaces LSTM with Bi-LSTM. The performance of the proposed method is better than that of the traditional GNN in [21], which shows that

Bi-LSTM has stronger ability to model temporal features than traditional LSTM.

3) An ablation study: to verify if the proposed method has the ability to capture spatiotemporal features from wind power data, an ablation study is conducted to analyze how each part (i. e., GCN and Bi-LSTM) of the proposed method works. The average CWCs of two datasets with 90% and 95% PINCs are visualized in Fig. 11.

It is clear that the proposed method based on the spatiotemporal features has a smaller CWC than others based on a single feature, implying that the proposed method is able to model spatiotemporal features from wind power data and meteorological factors accurately. For instance, for the first dataset with 90% PINC in summer, the CWC of the proposed method is reduced by approximately 12.10% compared with the Bi-LSTM that considers only temporal features. The CWC of the proposed method is reduced by 16.23% for the first dataset with 95% PINC in summer, indicating that the proposed method can portray temporal dependence of wind power data. Compared with GCN, which considers spatial features and ignores temporal features, for the first dataset with 90% PINC and 95% PINC in summer, the CWCs of the proposed method are decreased by approximately 6.03% and 21.95%, respectively, implying that the proposed method can portray spatial dependence well.

Further, the time complexity is tested by performing each

method 30 times. Specifically, the samples from the first dataset in spring are used as a simple example. The average training time (i.e., the time to train a model) and inference time (i.e., the time to obtain PIs of a sample using the trained model) of different methods are listed in Table VI.
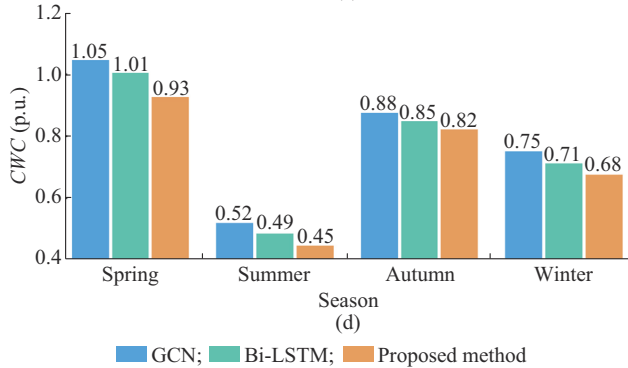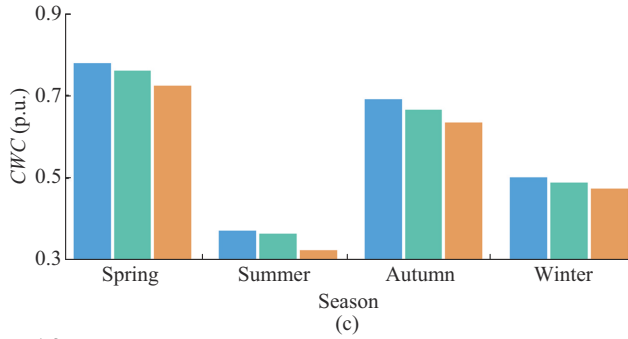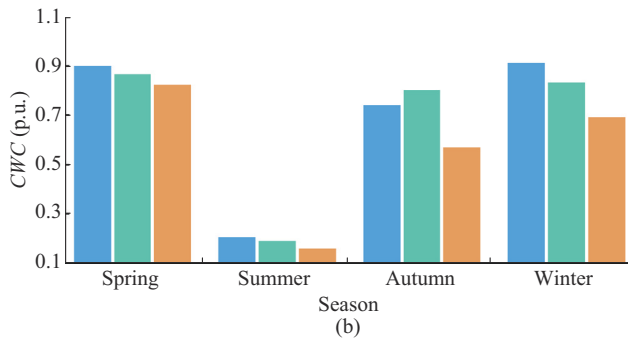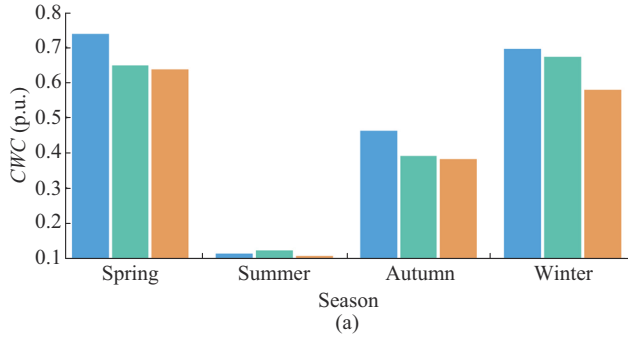

Fig. 11. Average CWCs of two datasets with different PINCs. (a) CWC of the first dataset with 90% PINC. (b) CWC of the first dataset with 95% PINC. (c) CWC of the second dataset with 90% PINC. (d) CWC of the second dataset with 95% PINC.

The training time of the proposed method is relatively long, which is the main disadvantage of the proposed method. It has to be mentioned that a few minutes of training time is acceptable in practical engineering. In addition, the inference time of each method is far less than 1 s, which can meet the real-time requirement of ultra-short-term prediction of wind power.

TABLE VI
AVERAGE TRAINING AND INFERENCE TIME OF DIFFERENT METHODS

Method	Training time (s)	Inference time (s)
MLP	14.836	0.002
LightGBM	2.783	0.001
GCN	29.514	0.003
Bi-LSTM	196.731	0.004
Proposed method	393.638	0.008
GNN	262.438	0.006

E. Interval Prediction with Different Time Horizons

In fact, the multi-step wind power interval prediction with one-hour-ahead and two-hour-ahead has been implemented to obtain satisfactory PIs based on the proposed method. Besides wind power interval prediction with the hourly horizon, the wind farm controller and transmission system operator are also highly interested in intra-hour PIs. For example, the 30-min measures are indispensable to reserve dispatch, continuous generation, wind farm control, and so on. In addition to the 1-hour time horizon in previous sections, this subsection further tests the performance of the proposed method for different look-ahead horizons, e.g., 0.5, 1.5, and 2 hours, and average prediction results of the test set with 95% PINC are given in Table VII.

From Table VII, it can be found that the CWC of each method increases as the prediction horizon becomes larger. This is because the uncertainty of wind power intensifies with the prediction horizon, which leads to a larger prediction error for deterministic point prediction models. To ensure sufficient PICPs, each method has to increase the width of PIs, which eventually results in a large CWC. Further, no matter how the time horizon changes, the proposed method can obtain the superior performance than other baselines (e.g., MLP, LightGBM, GCN, Bi-LSTM, and GNN) for hourly and intra-hourly wind power interval predictions. With successful application to the first dataset and second dataset in this paper, the proposed method can perform well for ultra-short-term wind power interval prediction no matter whether the dataset includes or excludes meteorological factors, indicating the proposed method is highly flexible for various datasets with different data compositions.

Practically, the wind farm controller and transmission system operator are likely to focus on system-level aggregated wind power. In this case, historical wind power of wind farms and surrounding meteorological factors can be taken as inputs to the proposed method with high flexibility to predict intervals of aggregated wind power based on the farm-level information. With the high precision and flexibility, the proposed method provides PIs of ultra-short-term wind power to facilitate various rise-based decision-making tasks (e.g., interval optimization and robust optimization of power systems) to determine the needed reserve [35].

TABLE VII
AVERAGE METRICS OF DIFFERENT PREDICTION METHODS WITH DIFFERENT TIME HORIZONS

Dataset	Season	Horizon (hour)	MLP			LightGBM			GCN			Bi-LSTM			GNN			Proposed method		
			PICP (%)	PINAW (p.u.)	CWC (p.u.)	PICP (%)	PINAW (p.u.)	CWC (p.u.)	PICP (%)	PINAW (p.u.)	CWC (p.u.)	PICP (%)	PINAW (p.u.)	CWC (p.u.)	PICP (%)	PINAW (p.u.)	CWC (p.u.)	PICP (%)	PINAW (p.u.)	CWC (p.u.)
First	Spring	0.5	90	0.345	0.791	94	0.336	0.690	90	0.296	0.670	92	0.285	0.614	92	0.274	0.597	93	0.268	0.569
		1.5	95	0.560	1.135	94	0.542	1.112	90	0.498	1.130	92	0.489	1.055	91	0.476	1.049	92	0.450	0.977
		2.0	95	0.652	1.320	95	0.630	1.272	88	0.572	1.368	93	0.565	1.202	90	0.527	1.200	91	0.511	1.149
	Summer	0.5	95	0.204	0.204	98	0.198	0.198	98	0.148	0.148	98	0.151	0.151	98	0.148	0.148	97	0.125	0.125
		1.5	94	0.435	0.895	97	0.355	0.355	98	0.252	0.252	98	0.237	0.237	97	0.229	0.229	97	0.196	0.196
		2.0	93	0.434	0.905	95	0.384	0.384	98	0.293	0.293	97	0.276	0.276	96	0.271	0.271	96	0.226	0.226
	Autumn	0.5	85	0.299	0.793	87	0.314	0.786	91	0.205	0.462	93	0.196	0.412	92	0.189	0.406	92	0.180	0.388
		1.5	94	0.817	1.685	86	0.428	1.104	89	0.456	1.061	89	0.425	0.997	88	0.387	0.936	87	0.349	0.870
		2.0	94	0.818	1.692	93	0.580	1.222	89	0.513	1.202	88	0.491	1.182	88	0.468	1.149	88	0.414	0.996
	Winter	0.5	95	0.367	0.738	92	0.285	0.610	92	0.302	0.645	91	0.228	0.503	92	0.221	0.479	92	0.212	0.454
		1.5	94	0.706	1.468	91	0.524	1.162	88	0.458	1.114	93	0.518	1.082	91	0.428	0.951	91	0.406	0.891
		2.0	94	0.732	1.499	93	0.628	1.335	88	0.535	1.311	94	0.624	1.290	89	0.543	1.269	91	0.550	1.227
Second	Spring	0.5	88	0.328	0.791	89	0.325	0.772	86	0.287	0.747	89	0.312	0.726	88	0.301	0.726	87	0.284	0.700
		1.5	89	0.575	1.339	89	0.573	1.359	85	0.496	1.336	86	0.523	1.338	86	0.497	1.280	86	0.470	1.220
		2.0	88	0.676	1.616	87	0.647	1.592	85	0.600	1.570	86	0.576	1.484	85	0.563	1.477	86	0.548	1.390
	Summer	0.5	95	0.294	0.596	99	0.394	0.394	99	0.375	0.375	98	0.373	0.373	98	0.355	0.355	98	0.313	0.313
		1.5	92	0.549	1.192	98	0.601	0.601	97	0.591	0.591	96	0.576	0.576	96	0.548	0.548	96	0.452	0.452
		2.0	93	0.639	1.359	97	0.668	0.668	96	0.659	0.659	95	0.624	0.624	95	0.614	0.614	95	0.596	0.596
	Autumn	0.5	93	0.312	0.659	92	0.301	0.645	92	0.300	0.646	88	0.259	0.619	91	0.269	0.591	93	0.274	0.579
		1.5	88	0.496	1.206	94	0.537	1.101	94	0.511	1.045	88	0.434	1.049	91	0.465	1.039	94	0.495	1.027
		2.0	94	0.635	1.298	89	0.536	1.279	90	0.563	1.282	95	0.619	1.252	93	0.593	1.252	93	0.596	1.240
	Winter	0.5	91	0.226	0.497	94	0.225	0.468	95	0.267	0.535	96	0.266	0.266	95	0.256	0.256	95	0.230	0.230
		1.5	95	0.484	0.971	93	0.464	0.974	95	0.481	0.972	92	0.439	0.959	93	0.431	0.915	94	0.435	0.884
		2.0	91	0.529	1.163	93	0.545	1.149	95	0.577	1.158	95	0.583	0.583	95	0.562	0.562	95	0.525	0.525

V. DISCUSSIONS

In this paper, the goal is to propose a new GNN and the improved Bootstrap technique for ultra-short-term interval prediction of wind power. The key factors affecting the performance of the proposed method are the standard deviations to be initialized in the improved Bootstrap technique.

Both theoretical analysis and simulation suggest that the parameter s_1 should be larger than the parameter s_2 to obtain a wide enough PIs, which can cover real values with a specified probability (i.e., PI nominal confidence). When the proposed method is migrated into other datasets, these key parameters can be determined by similar simulation steps in Section IV-B.

VI. CONCLUSION

To improve the precision of ultra-short-term prediction of wind power, this paper attempts to model the inputs as a graph from a new perspective. A GNN-based point prediction model is presented to model spatiotemporal features, and then an improved Bootstrap technique is proposed to obtain high-quality PIs. Through numerical simulation on two real-world datasets, the following conclusions are obtained.

1) The improved Bootstrap technique can effectively reduce the width of PIs with negligible reduction of PICP, especially for wind power generation curves with weak vola-

tile regions.

2) Compared with other popular point prediction methods (e.g., MLP, LightGBM, GCN, Bi-LSTM, and GNN in [21]), the proposed method has better precision for ultra-short-term interval prediction of wind power under different confidence levels and seasons, since it can capture spatiotemporal features from time-series data accurately.

3) No matter how the time horizon changes, the proposed method can obtain the superior performance to other baselines (e.g., MLP, LightGBM, GCN, Bi-LSTM, and GNN in [21]) for hourly and intra-hourly wind power interval predictions. Practically, the proposed method with a high precision and flexibility can provide high-quality PIs of ultra-short-term prediction of wind power to facilitate various rise-based decision-making tasks to determine needed reserves.

Although the numerical simulation results show that the proposed method outperforms popular baselines, it still has some limitations to be addressed.

1) The traditional and improved Bootstrap techniques have similar PIs for the strong volatile regions, since the reduction of the interval width of these regions easily causes the PICP to drop, and the improved Bootstrap technique only aims to reduce the width of PIs for the weak volatile regions. In future works, the Bootstrap technique can be further improved to target regions with strong volatility.

2) In addition to hourly and intra-hourly wind power inter-

val predictions, the proposed method may be extended to wind power prediction with a longer time horizon.

3) The widely-used PICP, PINAW, and CWC are used to test the performance of the proposed method. In the future, more metrics (e.g., pinball loss, Winkler score, and continuous ranked probability score) can also be used for further evaluation of models.

APPENDIX A

Algorithm 1: construction of PIs using improved Bootstrap technique

---

% Initialize parameters α, Bootstrap repeats Br, $s1$, $s2$:
Alpha=0.95; % $\alpha=0.95$;
Br=5000; % **Bootstrap repeats is 5000**
$s1$=0.036; % s_1=0.036;
$s2$=0.024; % s_2=0.024;
% Calculate prediction errors and standard deviation of validation set
Errors=Predictions-Real_values_of_validation_set
STD_Error=STD(Point_predictions_of_ validation_set)
% Assign errors to group 1 and group 2
k=1
for i=1:length(STD_Error)
  Group1(i)=Errors(i)
  if STD_Error<$s1$
    Group2(k)=Errors(i)
    k=k+1
  end
end
% Assign errors to group 3
if STD(Point_predictions_of_test_set)<$s2$
  Group3=Group2
else
  Group3=Group1
end
% Assign errors to group 4
for i=1:Br
  id=randperm(length(Group3))
  Group4(i)=Group3(id(1))
end
% Construct PIs
Group4 = sort(Group4);
Lower = percentile(Group4, (1 − Alpha)/2)
Upper=percentile(Group4, Alpha +((1 − Alpha)/2))
% Output results
sprintf('%.2f', Lower)
sprintf('%.2f', Upper)

REFERENCES

[1] J. Yan, Y. Liu, S. Han *et al*., "Reviews on uncertainty analysis of wind power forecasting," *Renewable and Sustainable Energy Reviews*, vol. 52, pp. 1322-1330, Dec. 2015.

[2] C. Li, G. Tang, X. Xue *et al*., "Short-term wind speed interval prediction based on ensemble GRU model," *IEEE Transactions on Sustainable Energy*, vol. 11, no. 3, pp. 1370-1380, Jul. 2020.

[3] Y. Zhou, Y. Sun, S. Wang *et al*., "Performance improvement of very short-term prediction intervals for regional wind power based on composite conditional nonlinear quantile regression," *Journal of Modern Power Systems and Clean Energy*, vol. 10, no. 1, pp. 60-70, Jan. 2022.

[4] Y. Wu, P. Su, T. Wu *et al*., "Probabilistic wind-power forecasting using weather ensemble models," *IEEE Transactions on Industry Applications*, vol. 54, no. 6, pp. 5609-5620, Nov.-Dec. 2018.

[5] T. Hong, P. Pinson, Y. Wang *et al*., "Energy forecasting: a review and outlook," *IEEE Open Access Journal of Power and Energy*, vol. 7, pp. 376-388, Oct. 2020.

[6] Y. Dong, S. Ma, H. Zhang *et al*., "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, Sept. 2022.

[7] J. Ding, K. Xie, B. Hu *et al*., "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, Sept. 2022.

[8] Z. Cui, X. Qing, H. Chai *et al*., "Real-time rainfall-runoff prediction using light gradient boosting machine coupled with singular spectrum analysis," *Journal of Hydrology*, vol. 603, pp. 1-15, Dec. 2021.

[9] Y. Wang, W. Liao, and Y. Chang, "Gated recurrent unit network-based short-term photovoltaic forecasting," *Energies*, vol. 11, no. 8, pp. 1-14, Aug. 2018.

[10] A. Khosravi, S. Nahavandi, and D. Creighton, "Construction of optimal prediction intervals for load forecasting problems," *IEEE Transactions on Power Systems*, vol. 25, no. 3, pp. 1496-1503, Aug. 2010.

[11] A. Khosravi, S. Nahavandi, D. Creighton *et al*., "Comprehensive review of neural network-based prediction intervals and new advances," *IEEE Transactions on Neural Networks*, vol. 22, no. 9, pp. 1341-1356, Sept. 2011.

[12] C. Wan, J. Lin, Y. Song *et al*., "Probabilistic forecasting of photovoltaic generation: an efficient statistical approach," *IEEE Transactions on Power Systems*, vol. 32, no. 3, pp. 2471-2472, May 2017.

[13] A. Khosravi, S. Nahavandi, D. Creighton *et al*., "Wind farm power uncertainty quantification using a mean-variance estimation method," in *Proceedings of IEEE International Conference on Power System Technology*, Auckland, New Zealand, Oct. 2012, pp. 1-6.

[14] A. Saeed, C. Li, M. Danish *et al*., "Hybrid bidirectional LSTM model for short-term wind speed interval prediction," *IEEE Access*, vol. 8, pp. 182283-182294, Sept. 2020.

[15] Y. Wen, D. AlHakeem, P. Mandal *et al*., "Performance evaluation of probabilistic methods based on bootstrap and quantile regression to quantify PV power point forecast uncertainty," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 4, pp. 1134-1144, Jun. 2019.

[16] Y. Li, X. Chen, C. Li *et al*., "A hybrid deep interval prediction model for wind speed forecasting," *IEEE Access*, vol. 9, pp. 7323-7335, Dec. 2021.

[17] K. Li, R. Wang, H. Lei *et al*., "Interval prediction of solar power using an improved Bootstrap method," *Solar Energy*, vol. 159, pp. 97-112, Jan. 2018.

[18] W. Liao, B. Bak-Jensen, J. R. Pillai *et al*., "Short-term power prediction for renewable energy using hybrid graph convolutional network and long short-term memory approach," *Electric Power Systems Research*, vol. 211, pp. 1-7, Oct. 2022.

[19] Y. Liu, Y. Liu and C. Yang, "Modulation recognition with graph convolutional network," *IEEE Wireless Communications Letters*, vol. 9, no. 5, pp. 624-627, May 2020.

[20] M. Yu, Z. Zhang, X. Li *et al*., "Superposition graph neural network for offshore wind power prediction," *Future Generation Computer Systems*, vol. 113, pp. 145-157, Dec 2020.

[21] 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, Apr. 2019.

[22] X. Geng, L. Xu, X. He *et al*., "Graph optimization neural network with spatio-temporal correlation learning for multi-node offshore wind speed forecasting," *Renewable Energy*, vol. 180, pp. 1014-1025, Dec. 2020.

[23] C. Draxl, A. Clifton, B. Hodge *et al*., "The wind integration national dataset (WIND) toolkit," *Applied Energy*, vol. 151, pp. 355-366, Aug. 2015.

[24] L. Zhao, Y. Song, C. Zhang *et al*., "T-GCN: a temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848-3858, Sept. 2020.

[25] W. Liao, B. Bak-Jensen, J. R. Pillai *et al*., "A review of graph neural networks and their applications in power systems," *Journal of Modern Power Systems and Clean Energy*, vol. 10, no. 2, pp. 345-360, Mar. 2022.

[26] Y. Yu, X. Si, C. Hu *et al*., "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235-1270, Jul. 2019.

[27] A. Khosravi, S. Nahavandi, D. Srinivasan *et al*., "Constructing optimal prediction intervals by using neural networks and bootstrap method," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 8, pp. 1810-1815, Aug. 2015.

[28] C. Li, G. Tang, X. Xue *et al*., "The short-term interval prediction of wind power using the deep learning model with gradient descend optimization," *Renewable Energy*, vol. 155, pp. 197-211, Aug. 2020.

[29] G. Sideratos and N. D. Hatziargyriou, "An advanced statistical method for wind power forecasting," *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 258-265, Feb. 2007.

[30] National Renewable Energy Laboratory. (2022, Sept.). Geospatial data science applications and visualizations. [Online]. Available: https://maps.nrel.gov/wind-prospector/

[31] S. Samadianfard, S. Hashemi, K. Kargar *et al*., "Wind speed prediction using a hybrid model of the multi-layer perceptron and whale optimization algorithm," *Energy Reports*, vol. 6, pp. 1147-1159, Nov. 2020.

[32] W. Liao, D. Yang, Y. Wang *et al*., "Fault diagnosis of power transformers using graph convolutional network," *CSEE Journal of Power and Energy Systems*, vol. 7, no. 2, pp. 241-249, Mar. 2021.

[33] W. Liao, Z. Yang, X. Chen *et al*., "WindGMMN: scenario forecasting for wind power using generative moment matching networks," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 5, pp. 843-850, Oct. 2022.

[34] W. Liao, B. Bak-Jensen, J. R. Pillai *et al*., "Scenario generations for renewable energy sources and loads based on implicit maximum likelihood estimations," *Journal of Modern Power Systems and Clean Energy*, vol. 10, no. 6, 1563-1575, Nov. 2022.

[35] Z. Yang, W. Liao, Q. Zhang *et al*., "Fault Coordination control for converter-interfaced sources compatible with distance protection during asymmetrical faults," *IEEE Transactions on Industrial Electronics*, doi: 10.1109/TIE.2022.3204946

Wenlong Liao received the B.S. degree from China Agricultural University, Beijing, China, in 2017. He received the M.S. degree from Tianjin University, Tianjin, China, in 2020. He is currently pursuing the Ph.D. degree in Aalborg University, Aalborg, Denmark. His current research interests include smart grids, machine learning, and renewable energy.

Shouxiang Wang received the B.S. and M.S. degrees in electrical engineering from Shandong University of Technology, Jinan, China, in 1995 and 1998, respectively, and the Ph.D. degree in electrical engineering from Tianjin University, Tianjin, China, in 2001. He is currently a Professor with the School of Electrical and Information Engineering, and Deputy Director of Key Laboratory of Smart Grid of Ministry of Education, Tianjin University. His main research interests are distributed generation, microgrid, and smart distribution system.

Birgitte Bak-Jensen received the M. Sc. degree in electrical engineering and the Ph.D. degree in modeling of high voltage components from the Institute of Energy Technology, Aalborg University, Aalborg, Denmark, in 1986 and 1992, respectively. She is currently a Professor with the Department of Energy Technology, Aalborg University. Her current research interests include distribution system analysis, smart grids, and intelligent energy systems.

Jayakrishnan Radhakrishna Pillai received the M. Tech. degree in power systems from the National Institute of Technology, Calicut, India, in 2005, the M.Sc. degree in sustainable energy systems from the University of Edinburgh, Edinburgh, U.K., in 2007, and the Ph.D. degree in power systems from Aalborg University, Aalborg, Denmark, in 2011. He is currently an Associate Professor with the Department of Energy Technology, Aalborg University. His current research interests include distribution system analysis, grid integration of electric vehicles and distributed energy resources, smart grids, and intelligent energy systems.

Zhe Yang received the B.S. degree from Northeast Electric Power University, Jilin, China, in 2017. He received the M. S. degree from North China Electric Power University, Beijing, China, in 2020. He is currently pursuing the Ph.D. degree in Aalborg University, Aalborg, Denmark. His current research interests include machine learning, renewable energy, and power system protection.

Kuangpu Liu received the B.S. and M.S. degree from Northwestern Polytechnical University, Xi'an, China, in 2017 and 2020, respectively. He is currently pursuing the Ph. D. degree in Aalborg University, Aalborg, Denmark. His current research interests include reinforcement learning, machine learning, and renewable energy.