Direct Interval Forecast of Uncertain Wind Power Based on Recurrent Neural Networks

Zhichao Shi, *Student Member, IEEE*, Hao Liang, *Member, IEEE*, and Venkata Dinavahi, *Senior Member, IEEE*

***Abstract*—Interval forecast is an efficient method to quantify the uncertainties in renewable energy production. In this paper, the idea of prediction intervals (PIs) is employed to capture the uncertainty of wind power generation in power systems. The recurrent neural network (RNN) model is proposed to construct PIs with the lower upper bound estimation method, which is a powerful nonparametric forecast approach. In addition, a novel comprehensive cost function with a new PI evaluation index is designed with the purpose of enhancing the model training. To tune the parameters of RNN prediction model, the dragonfly algorithm with a linearly random weight update method is introduced as the optimization tool. The performance of the proposed prediction model is validated by a case study using a real world wind power dataset, and the comparative results show the superiority of the model.**

***Index Terms*—Lower upper bound estimation (LUBE), optimization, recurrent neural network (RNN), wind power prediction.**

I. Introduction

With the development of advanced generation technologies, there has been an enormous increase in the amount of renewable energy generation in recent years. As a clean and widely available resource, wind energy has become one of the most popular renewable sources [1]. The installed capacity in Canada grew by an annual rate of 18% in the past five years and it exceeded 10,000 MW in 2015 [2]. However, wind power is also intermittent and uncertain resulting from the volatile wind speed. Consequently, increasing penetration of wind power generation in the future smart grid will pose new challenges to system operation and dispatch. In order to deal with variable wind power, different methods have been proposed to forecast it aimed at reducing its uncertain impact on power system operation.

Manuscript received June 14, 2017; revised August 29, 2017 and October 6, 2017; accepted November 13, 2017. Date of publication November 16, 2017; date of current version June 18, 2018.This work was supported in part by the Natural Science and Engineering Research Council of Canada and in part by Grant 201603170278. The work of Z. Shi was supported by the China Scholarship Council under Grant 201603170278. *(Corresponding author: Zhichao Shi.)*

Z. Shi is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada, and also with the College of Information System and Management, National University of Defense Technology, Changsha 410073, China. (e-mail: zhichao1@ualberta.ca).

H. Liang and V. Dinavahi are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada (e-mail: hao2@ualberta.ca; dinavahi@ualberta.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSTE.2017.2774195

Wind power forecast consists of direct wind power forecast and indirect wind speed forecast. Based on the forecast time scale, wind power prediction can be classified into short-term, medium-term and long-term prediction [3]. While according to the forecast models, it can be divided into two main categories: physical and statistical [4]. Physical methods are developed by modeling the relationship between physical variables and the specifications of wind turbines with some physical-based equations. For the statistical methods, time series models (e.g., autoregressive integrated moving average ARIMA) [5], data mining approaches [6], artificial neural networks (ANNs) [7], [8], and support vector machines (SVM) [9] are widely studied in the literature. Compared with physical models, statistical models, as a data-driven technique, are usually simpler and more adaptive. In addition, hybrid methods which combine different models have also been researched [10] to improve the forecast performance. This kind of approach aims to retain the advantages of the individual model and it seems to be better than other methods.

Conventional point forecast methods as mentioned above suffer from the problem that they cannot eliminate the forecast error and the forecast accuracy may be highly variable. They only generate a deterministic forecast value for a certain time step without any indication of the associated uncertainty [11]. From the decision making point of view, the use of point forecast will have significant impact on the stable and reliable operation of power system. Therefore, recent research of wind power prediction have focused more on probabilistic methods which can include the forecast uncertainty [12].

Compared with the widely used point prediction methods, probabilistic forecasts could provide additional quantitative information about wind power generation uncertainty [13]. As a result, the probabilistic forecast of wind energy has attracted much attention recently [14], [15]. In probabilistic forecast, the uncertainty can be expressed by probabilistic measures such as probability density functions (PDFs), quantiles and intervals, moments of distribution (mean and variance) and so on [13]. As the most visualized representation, interval forecast of wind energy gained more popularity [16], [17].

Among different kinds of interval forecast approaches, the recently proposed lower upper bound estimation (LUBE) method is a nonparametric procedure that can construct appropriate prediction intervals (PIs) directly in an unsupervised learning mode based on a feedforward neural network (NN) [18]. It makes no assumption about the forecast error distribution and its computational burden is almost negligible in comparison with other

traditional NN-based PI construction methods [19]. To obtain high quality PIs with narrow width and large coverage probability, both single-objective procedure [20] and multi-objective framework [21] using LUBE method are proposed in the literature. In [22], a constrained single-objective framework is designed to minimize the PI width while constraining the coverage probability. Similarly, a hybrid model based on back propagation NN is proposed for wind speed interval forecast [23]. In this study, the wavelet de-noising technique is employed to preprocess the data, and cuckoo search optimization algorithm is used to train the NN model. In the multi-objective framework, Pareto optimal solutions [21] and fuzzy inference method [24] are investigated to construct the optimal PIs. In these studies, the prediction models are all based on the feedforward NN model. In addition, rough neural networks which combine rough and conventional neurons to deal with uncertainties of the data are also studied for wind power or speed prediction [25]. In [26], the rough neurons are integrated in a deep neural architecture to improve the accuracy of short-term wind speed forecast. In [27], the most informative input parameters of an NN are determined through attribute reduction of rough set theory and the training time is also reduced. Although good results are reported in these works, they concentrate on wind speed point forecast instead of interval forecast. The combination of deterministic and probabilistic forecast for wind power is also studied [28]. It is an indirect probabilistic forecast method which firstly implements point forecast with support vector regression (SVR) models and then the confidence intervals for forecast values are obtained with quantile regression (QR) method.

Although NN-based LUBE method has been widely studied as aforementioned, the existing research, to our best knowledge, focus on the feedforward NN prediction model, especially the multilayer perceptron (MLP) NN model. Generally, NN models have two basic structures: feedforward and feedback. Compared with the feedforward NN, recurrent neural network (RNN) with a feedback structure has been shown to excel at time series forecast [29]. In a feedback structure, the network behaves like a dynamic system which can better capture the characteristics of variable wind speed [30]. In the last decades, the RNN model has been extensively studied for wind power and speed forecast. For instance, the local RNNs are studied for the long-term wind speed and power prediction in [31]. These models were trained by two on-line learning methods based on recursive forecast error algorithm. Similarly, the long-term wind speed prediction was studied using a composite method of statistical and NN approaches in [3]. Based on the general trend and pattern extracted from the historical data, the nonlinear autoregressive with exogenous inputs (NARX) network was trained and used to forecast the next year hourly data. Moreover, the Elman neural network was also explored for wind speed forecast [32]. Despite considerable research on RNN, these works focus on point prediction based on forecast error. However, interval prediction with RNN model is rarely reported in the literature.

In this study, the short-term wind power interval prediction based on RNN model and LUBE method is investigated for the first time. We employ the single-objective framework and a novel aggregated cost function is designed as the objective of model training. Considering the high complexity and nonlinearity of the cost function, the dragonfly algorithm (DA)-a new intelligent and powerful optimization algorithm-is introduced to solve the problem effectively. In addition, a new modification is proposed for DA to reinforce its search ability. To cope with the chaotic historical wind power data, delay embedding theorem instead of the general correlation analysis is applied in this study. The proposed RNN interval prediction model has been evaluated by the case study with a real world wind power dataset.

In summary, the main contributions of this work are as follows:

1) The RNN model exhibiting the dynamic system behaviors is investigated for wind power interval forecast; this is the first time to apply RNN model to do interval forecast for wind power;
2) A new evaluation index for the PI width is proposed to enhance the RNN model training, this new index, unlike the previous measures, further considers and uses the known information in the training process;
3) DA is introduced for the first time to solve the PI problem and a new weight update method which combines linear decrease and random walk is designed to improve the algorithm search ability;
4) Delay embedding approach rather than the typical correlation analysis is employed to reconstruct the time series data and determine the input of the prediction model. It is suitable to process wind power data with chaotic characteristics.

The remainder of this paper is organized as follows. The background of PI is introduced in Section II. In Section III, the proposed RNN interval prediction model and the DA algorithm are explained. The wind energy prediction case study is implemented in Section IV. Finally, Section V concludes this paper with some remarks for future study.

II. Background of PI

The construction of a PI is to estimate the upper and lower bound of an interval with a confidence level which shows the accuracy and the quality of PIs need to be evaluated by some measures. In this section, the PI evaluation indices are introduced first. Then, the NN-based LUBE interval prediction method is explained.

A. PI Evaluation Indices

A high quality PI is expected to have larger reliability and narrower width. To assess these two aspects of PIs, two indices, PI coverage probability (PICP) and PI normalized average width (PINAW) [19], are mostly employed to quantitatively measure the forecast intervals. PICP which is also called PI confidence level is used to show the probability that target values will be covered by the forecasted intervals. Obviously, a larger PICP value indicates that more targets will lie in the constructed PIs. This index is usually considered as the critical indicator of PIs

and it can be mathematically defined as follows:

$$\text{PICP} = \frac{1}{N}\sum_{i=1}^{N}\delta_i \tag{1}$$

where N is the number of test samples and δ_i is a binary value which is described as follows:

$$\delta_i = \begin{cases} 1, & y_i \in [L_i, U_i] \\ 0, & y_i \notin [L_i, U_i] \end{cases} \tag{2}$$

where y_i is the target, L_i and U_i are lower and upper bound of the PI, respectively. Generally, the PICP value should be greater than the preassigned confidence level in the training process, otherwise the PIs are invalid and should be discarded.

Although PICP index is the key feature of PIs, we can not just focus on this objective and ignore the interval width. With sufficiently wide intervals, the PICP objective can be easily achieved. However, very wide intervals hardly yield any valuable information and may be useless in practice. Therefore, a quantitative measure of interval width, PINAW, is defined to limit the interval width, as follows:

$$\text{PINAW} = \frac{1}{N \cdot Rg}\sum_{i=1}^{N}(U_i - L_i) \tag{3}$$

where Rg means the range of the targets (difference between the maximum and the minimum).

PICP and PINAW can only evaluate one of the aspects of PIs, respectively. In order to assess the overall quality of PIs, the index coverage width-based criterion (CWC) which is a comprehensive cost function is designed [20]. Furthermore, a modified CWC cost function is proposed to overcome the multiplication drawback [33] as follows:

$$\text{CWC} = \text{PINAW} + \gamma(\text{PICP})e^{-\eta(\text{PICP}-\mu)} \tag{4}$$

where $\gamma(\text{PICP})$ is a boolean function given by

$$\gamma = \begin{cases} 1, & \text{if PICP} < \mu \\ 0, & \text{if PICP} \geq \mu \end{cases} \tag{5}$$

where η and μ are two controlling parameters. The former is usually a large constant to penalize the invalid PIs, while the latter can be determined according to the nominal confidence level.

With the CWC function, the primary multi-objective problem can be transformed into a single-objective minimization problem. Although the CWC cost function is used as a comprehensive evaluation index, we can find that it is only determined by the estimated upper and lower bounds and the known information is not fully used in the training process. Similar to the frequently used MSE index in point forecast, a new PI width evaluation criterion, PIMSE, is designed to make better use of the known target values in this study:

$$\text{PIMSE} = \frac{1}{N}\sum_{i=1}^{N}[(U_i - y_i)^2 + (L_i - y_i)^2)]. \tag{6}$$

By introducing this index, a new CWC (NCWC) function can be developed for the training:

$$\text{NCWC} = \text{PINAW} + \gamma(\text{PICP})e^{-\eta(\text{PICP}-\mu)} + \text{PIMSE}. \tag{7}$$

With this new cost function, we can combine the unsupervised learning and supervised learning by using the known information in the training process. Besides, it is expected that a more symmetric interval, which is closer to the true confidence interval, will be obtained by minimizing the PIMSE index. Therefore, the NCWC is used to enhance the model training in this work.

In the model training process with the NCWC objective, the reliability index PICP will be the influential factor at the beginning stage due to the high penalty cost. If PICP is less than the predefined confidence level, the NCWC will be large regardless of the interval width. As the training continues, the PICP will become greater than the nominal confidence level and the sum of PINAW and PIMSE should be the dominant factor. Note that it is necessary to consider both criteria PINAW and PIMSE here as PINAW mainly focuses on narrower intervals and PIMSE will make full use of the known information and make the intervals more close to the actual symmetric confidence intervals. If only PIMSE index is considered, we may get symmetric but wide intervals which will lead to a large PINAW value.

B. LUBE Method

The LUBE method is a nonparametric method that can directly construct PIs [18]. It is simple and fast to generate PIs without any assumption about the forecast errors. In previous work, LUBE method is implemented with a feedforward NN. The NN model has two output nodes including the upper and lower bounds of PIs. Actually, this method belongs to unsupervised learning since the upper and lower bounds are not known during the training process. In practice, it is better to have narrower PIs with a larger coverage probability. Therefore, the primary problem based on LUBE method is a multi-objective problem with two conflicting objectives. This multi-objective problem can be transformed into a single-objective problem by introducing the nonlinear and nondifferentiable CWC cost function and some basic constraints [18], [22]. To optimize the cost function in the NN training process, different gradient-free optimization methods such as simulated annealing (SA), particle swarm optimization (PSO) algorithm have been applied in previous study.

In each iteration of the training process, two outputs representing the lower and upper bounds of a PI are generated based on the model inputs. Then the two measures PICP and PINAW as well as the corresponding CWC cost function can be calculated for all training samples. As the training procedure continues with a certain optimization algorithm, the NN parameters are tuned gradually so that the PICP meets the predefined confidence level and the interval width PINAW decreases. When the maximum number of iterations is reached or there is no further improvement on the objective for a certain number of consecutive iterations, the model training terminates and the resulting optimal model can be used for construction of new PIs [34]. The key features of LUBE method includes simplicity,

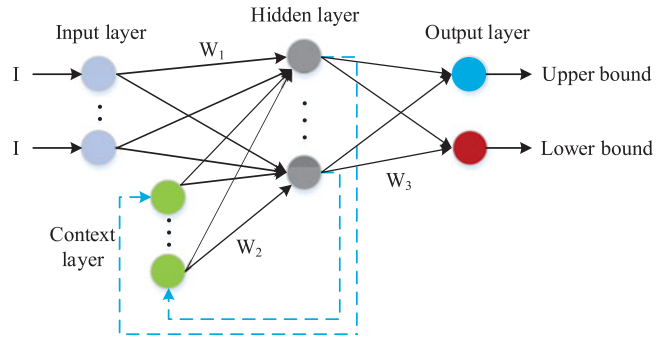

Fig. 1. Schematic diagram of Elman network.

low computation cost and distribution-free assumption compared with traditional interval forecast methods. More details about the LUBE method can be found in [18].

III. RNN-Based Interval Prediction Model

As mentioned above, the LUBE method is a simple and efficient method to construct high quality PIs. Due to the easy implementation and low computational cost, LUBE method has become popular in quantifying forecast uncertainty in a very short time. It has attracted much attention and abundant research works are carried out based on feedforward NN model to do interval prediction. Compared with the feedforward NN which can only learn a static input-output mapping relationship, an RNN with feedback structure behaves like a dynamic system which is more suitable to model temporal sequences. Therefore, we will develop the RNN-based interval prediction model for wind power generation in this study. Two typical RNN models, Elman network and NARX model, are investigated.

A. Elman Network

As a first-order RNN, the Elman network is a simple recurrent network [35] which employs a context layer to feedback the outputs of the hidden layer. The context layer is a copy of the hidden layer outputs at the previous time step. Although simple in structure, it has the ability to perform complex tasks. Based on the Elman network, the interval prediction model can be developed and the schematic diagram of the model is shown in Fig. 1.

In this three layer Elman network model, there are two output nodes representing the upper and lower bounds of a PI. The only feedback is from the hidden layer to the context layer and the connection weights are constants, while the other feedforward weights are adjustable which could be optimized by intelligent optimization method. According to the Elman structure, the dynamic change of this model can be mathematically expressed as follows:

$$x(k) = \phi[W_1 I(k-1) + W_2 x_c(k) + b_1] \tag{8}$$

$$x_c(k) = x(k-1) \tag{9}$$

$$z(k) = f(W_3 x(k) + b_2) \tag{10}$$

where x and z represent the output of the hidden layer and output layer, respectively, x_c is the output of the context layer, $\phi(\cdot)$ is the transfer function which is usually nonlinear hyperbolic-tangent-sigmoid function, and $f(\cdot)$ is pure linear activation function.

B. NARX Network

NARX is another class of RNN that is suitable to model time series data. Unlike Elman network, the feedback in this input-output recurrent model is from the output to the input. The dynamic behaviour of NARX model can be mathematically described by [36]:

$$o(k+1) = g[o(k), ..., o(k-d_o); u(k), ..., u(k-d_u)] \tag{11}$$

where o is the output, u is the input, d_o and d_u are the output and input delay, respectively, $g(\cdot)$ is also a nonlinear mapping function which can be approximated by a standard MLP model. Without loss of generality, the typical three layer structure is used in this study. Thus, the NARX-based interval forecast model is similar to the previously described Elman forecast model and the schematic diagram is omitted.

C. Dragonfly Algorithm

Inspired by the behaviours of dragonflies, DA is a population-based optimization algorithm that was proposed in 2016 [37]. In DA, each dragonfly represents a promising solution for the optimization problem. In this study, the dragonfly or solution is a vector that consists of the NN connection weight values. These weights are adjusted to find the optimal values by minimizing the cost function. To solve the problems with DA, each dragonfly should have two vectors: position (P) and step (V). The step vector here is similar with the velocity vector in PSO, while the position updating of individuals is determined by five main behaviours including separation, alignment, cohesion, attraction towards food and distraction from an enemy. These behaviours can be described as follows [37]:

$$Se_i = -\sum_{j=1}^{Num} P_j - P_i \tag{12}$$

$$Al_i = \left(\sum_{j=1}^{Num} V_j\right) / Num \tag{13}$$

$$Co_i = \left(\sum_{j=1}^{Num} P_j\right) / Num - P_i \tag{14}$$

$$Fo_i = P^+ - P_i, En_i = P^- + P_i \tag{15}$$

where P_i is the position of current individual, P_j and V_j represent the j-th neighbour individual's position and corresponding velocity, respectively, *Num* is the number of neighbour individuals, P^+ and P^- denotes the position of dragonflies' food and enemy, respectively.

Based on the above behaviours of the dragonflies, the position vector of each individual can be updated as follows:

$$P^{iter+1} = P^{iter} + V^{iter+1} \tag{16}$$

$$V^{iter+1} = (sSe_i + aAl_i + cCo_i + fFo_i + eEn_i) + wV^{iter} \tag{17}$$

where *iter* is the iteration number, *s*, *a*, *c*, *f*, *e*, and *w* are corresponding weight coefficients which control the exploration and exploitation search during the optimization process. The above updating rules are applicable for individuals with neighbours. When the individuals have no neighbours, the Levy flight which is a random walk [38] is used to improve their exploration and stochastic behaviours, and the position update is as follows:

$$P^{iter+1} = P^{iter} + \text{Levy}(dim) \times P^{iter} \tag{18}$$

$$\text{Levy}(dim) = 0.01 \times \frac{c_1 \times \rho}{|c_2|^{\frac{1}{\lambda}}} \tag{19}$$

where *dim* is the dimension of the vector P, c_1 and c_2 are two random numbers in [0, 1], respectively, λ is a constant value and ρ can be calculated by the following equation:

$$\rho = \left(\frac{\Gamma(1+\lambda) \times \sin(\frac{\pi\lambda}{2})}{\Gamma(\frac{1+\lambda}{2}) \times \lambda \times 2^{\frac{\lambda-1}{2}}} \right)^{1/\lambda} \tag{20}$$

where $\Gamma(n) = (n-1)!$.

The DA has been demonstrated to perform better than other well-known optimization algorithm such as PSO and genetic algorithm (GA) on the test functions [37]. Therefore, it is introduced to tune the RNN parameters by optimizing the comprehensive cost function in this study. In DA, the inertia weight *w* is adjusted adaptively by the typical linearly decreasing manner. In order to enhance the total search ability of this algorithm, inspired by the random inertia weight update method [39], we can further improve the weight update by using Levy flight as follows:

$$w^{iter} = w_{\max} - (iter/Maxt) \times (w_{\max} - w_{\min}) \tag{21}$$

$$w^{iter+1} = w^{iter} + Levy(dim) \tag{22}$$

where $w_{\max}$ and $w_{\min}$ are the maximum and minimum weights, respectively, *Maxt* is the maximum iteration number, the Levy function is the same as that defined in (20).

D. Model Implementation

Based on the Elman and NARX network model, the LUBE method was implemented to construct PIs with DA optimization algorithm. The single-objective problem is formulated with the NCWC cost function. The model implementation flowchart is shown in Fig. 2 and details of the main steps are discussed below.

1) Dataset partition and preprocess. For the forecast model, the main input is the historical wind power data. The original dataset should be split into training data, validation data and test data. The training and validation data are combined to train the model, while the test data are used to verify the model's generalization ability. In order to accelerate the model training process, the original data are usually normalized to $[-1, 1]$ after partition.

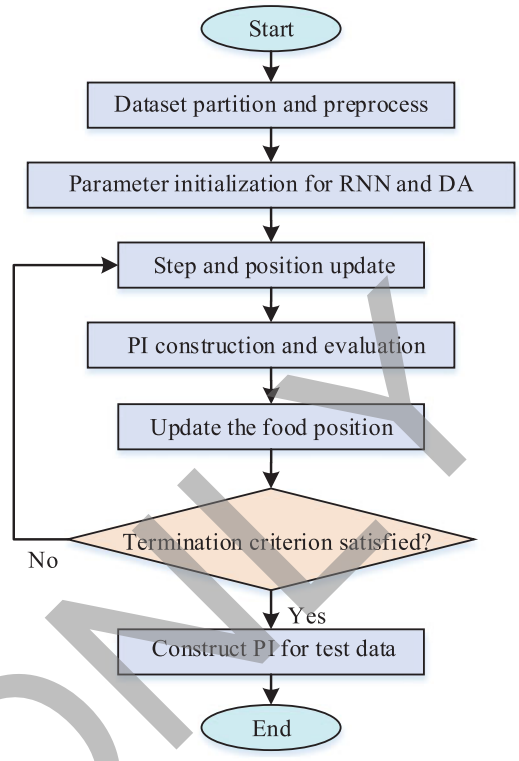

Fig. 2. Model flowchart for PI construction.

To construct PIs with the RNN model, the original time series data should be transformed into a suitable form for training the model. A dynamic system in discrete time can be depicted as:

$$X(t+1) = \mathbf{F}(X(t)) \tag{23}$$

where $X(t)$ is the system state at time step *t* and **F** is nonlinear vector valued function. Since the wind power and speed data are volatile and chaotic [40] from a dynamic system, the state space reconstruction technique with the delay embedding theorem [41] was employed to process the original data. By this theorem, the one-dimensional chaotic data is supposed to compress the information of higher dimension. Hence, the time series data $X(t)$ can be reconstructed as follows:

$$X_0(t) = [X(t), X(t-\tau), \ldots, X(t-(m-1)\tau)] \tag{24}$$

where τ is time delay and m is the embedding dimension. Therefore, creating the delay embedding comes down to finding the optimal values of parameters τ and m. For a given dataset, these two parameters can be determined by the mutual information function and false nearest neighbour method [42].

Once the delay embedding was completed, the reconstructed time series was generated and it was used to train the RNN model for one-step ahead prediction task. In this case, the number of model input was also determined which equals the embedding dimension.

2) Parameter initialization. There are two sets of parameters corresponding to the DA algorithm and RNN model. In DA

algorithm, the step and position vectors are initialized with small random numbers. The connection weights of RNN model are represented by the position vectors of dragonflies, thus their initialization is finished. To find the optimal number of hidden nodes in RNN model, five-fold cross-validation method can be applied with the training dataset.

3) Update step and position vectors. The position and step of each dragonfly are updated according to (16) and (17). The individual with better fitness (smaller cost function value in this study) will be retained.

4) Model evaluation. Each individual corresponds to one prediction model. With the connection weights, PIs can be constructed and the corresponding measures PICP, PINAW and PIMSE can be calculated. The index NCWC is considered as the fitness in RNN training process. The individual with the best fitness is recorded as dragonflies' food source and its position vector represents the best model weights.

5) Termination criterion. The training is terminated when the maximum iteration is reached in this work. If the termination condition is not met, then it will return to update the step and position vectors.

6) PI construction for test dataset. When the training is completed, we can get the optimal connection weights for RNN model. With this optimal prediction model, PI construction can be easily accomplished for the test data. The relevant indices are also calculated to evaluate the PI quality.

IV. Realistic Wind Energy Prediction Case Study

To validate the forecast performance of the proposed RNN-based LUBE method, a realistic case study of wind energy interval prediction is carried out in this section. First of all, the data source and relevant parameter settings are explained. Then, the numerical results and discussion are presented.

A. Data Set

The historical wind power data is taken from the Adelaide wind farm located in Ontario, Canada and it can be obtained from the IESO website [43]. Ontario takes the lead in the clean wind power utilization with 4781 MW of installed capacity, which supplies about 5% of the total electricity demand in the province [44].

The chosen dataset consists of hourly wind power data in MW from 1 January 2016 to 7 April 2017. During this time period, the wind farm performance is assumed to be normal and there is no missing or false data. The whole dataset is further partitioned into three subsets for training, validation and test. The training set and validation set account for about 80% of the whole dataset in this study, i.e., the whole year data in 2016. The remaining data are used to test the model's prediction performance.

B. Parameter Settings

As mentioned above, two sets of parameters about RNN model and DA algorithm should be determined in the proposed prediction model. For the RNN model, we should design the best structure by finding the optimal number of input nodes and hidden nodes. The number of input nodes is related to the dimension of delay vectors which can be determined by state space reconstruction technique. According to the delay embedding approach, a time series is a series of observations of a dynamic system and the forecast is about forecasting the system's state. To forecast the system, we should construct a state space that is equivalent to the original one by using a small set of the most recent previous observations [41]. By delay embedding theorem, we need to find two parameters: the embedding dimension m which represents the size of the set of most recent observations and the time delay τ which means the optimal autocorrelation level in each delay vector. In this study, τ is determined to be 16 by the mutual information method where the first local minimum occurs and m is 7 by false nearest neighbour method, which can be accomplished by the utility functions $mutual$ and $false\_nearest$ in TISEAN toolbox [45]. In this case, the dimension of reconstructed delay vectors is 7 and the number of input nodes for the RNN model is also 7. After determining the value of τ and m, delay vectors can also be obtained for the prediction model.

TABLE I
Cross-Validation Results of Elman Model

Nodes	PICP(%)	PINAW(%)	PIMSE	NCWC	CWC(%)
3	94.88	74.29	1.8427	2.5856	74.29
5	94.94	65.66	1.5107	**2.1672**	65.66
8	93.90	69.04	1.6974	2.3878	69.04
10	94.13	73.69	2.0186	2.7555	73.69

As for the number of hidden layer nodes, five-fold cross-validation method is employed to explore it with the training dataset in this work. This method is frequently used in the literature [18] which can help establish a stable model. The cross-validation results of some typical hidden nodes numbers for Elman network are shown in Table I. From this table, we can see that the model has the best performance according to the NCWC index when the number of hidden nodes is 5. Therefore, the optimal structure of Elman network is 7-5-2 and the number of nodes in the context layer is also 5. The optimal structure of NARX model can be determined similarly and the number of hidden nodes is also 5 in this study. In addition, for the typical three layer structure, we use the common hyperbolic tangent and pure linear activation functions in the hidden and output layer, respectively.

Another parameter set is about the optimization algorithm. It is suggested that the weight coefficients in (17) can update in an adaptively tuning method to balance the exploration and exploitation during the optimization [37]. Generally, the inertia weight w varies from 0.9 to 0.4 [46]. In this study, its range is set to be [0.7, 1] by trial and error method. In the NCWC or CWC cost function, the controlling parameter μ is set to be the nominal confidence level 0.9 and η equals 50 [11]. In addition, the population size is 30 and the maximum number of iteration is set to be 1000 during the optimization process.

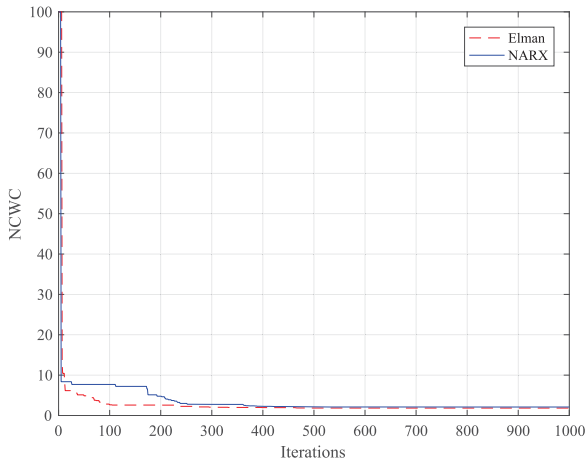

Fig. 3. NCWC of the best individual in the training process.

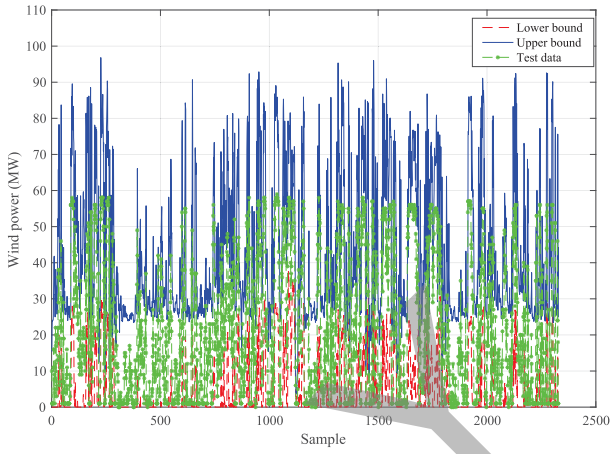

Fig. 4. Optimal PIs of Elman model for test data.

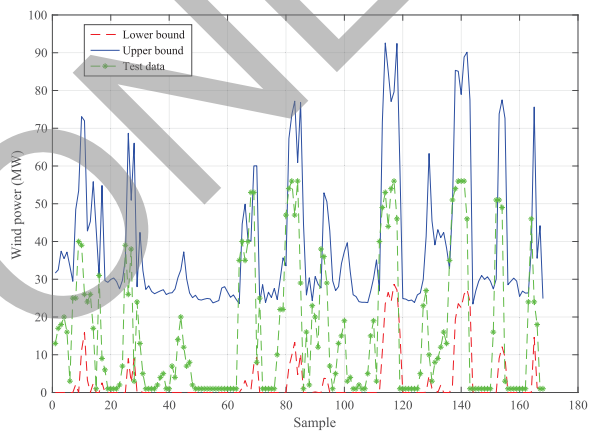

Fig. 5. Optimal PIs of Elman model for the last week of test data.

C. Test Results

The proposed Elman network and NARX interval prediction model have been applied to construct wind power PIs. After the training process, the dragonflies' food position vector obtained by DA corresponds to the optimal prediction model. With the best model, PIs can be constructed for the wind power test dataset. The performance measures are also calculated to evaluate the obtained PIs' quality including PICP, PINAW, PIMSE, NCWC and the CWC cost function which is frequently used in the literature.

In the training process, the variation of the best individual's objective function is shown in Fig. 3. As shown in this figure, the NCWC function of both Elman and NARX model decreases dramatically in the first few generations to get the satisfied PICP. When PICP satisfies the nominal coverage probability, more attentions are paid to the interval width, i.e., to minimize the sum of PINAW and PIMSE. As the optimization proceeds, the NCWC objective continues to decrease and it converges to the optimal value eventually for both models. The convergence validates the strong search ability of DA algorithm with random linear weight update method. It can also be seen that the optimal objective value of Elman network is a little better than that of NARX model.

For Elman network model, the PI construction results for test data are shown in Fig. 4. For better visualization, the PIs for the last week of test data are also given in Fig. 5. From the results, we can see that most of the target values (the green dash line with star) lie in the constructed upper and lower bounds. As shown in Fig. 5, both the predicted upper and lower bounds have a similar trend with the real data, which implies that the prediction model can capture the dynamic feature of wind power data well. Note that the lower bound that is below zero is set to be zero in this study. In addition to the similar trend of those three lines, we can also find that most of the constructed intervals are approximately symmetric about the targets resulting from the involvement of PIMSE in the training process. A symmetric interval is more closer to the true confidence interval which can be obtained based on the known distribution information.

TABLE II
PI CONSTRUCTION RESULTS OF ELMAN MODEL

No.	PICP(%)	PINAW(%)	PIMSE	NCWC	CWC(%)
1	93.86	61.95	1.2362	1.8558	61.95
2	93.90	65.20	1.3421	1.9940	65.20
3	93.81	63.71	1.4188	2.0559	**63.71**
4	93.94	66.43	1.3564	2.0207	66.43
5	94.24	62.90	1.3364	1.9654	62.90

In order to verify the repeatability of the prediction model and get convincing forecast results, the case study with Elman network is repeated for five times. Results of each run including PICP, PINAW, PIMSE, NCWC and CWC are shown in Table II. As can be seen from this table, the PICP values of all five runs satisfy the nominal coverage probability (90%), which indicates that the prediction model is reliable since the PICP index is usually considered as the key feature of the PIs. The results are also consistent as the variances of these measures are quite small, e.g., the standard deviation of CWC for five runs in Table II is 1.7912. Note that the PIMSE value here is calculated from the normalized data. As the CWC cost function is a comprehensive index which is frequently used in the literature, we will take

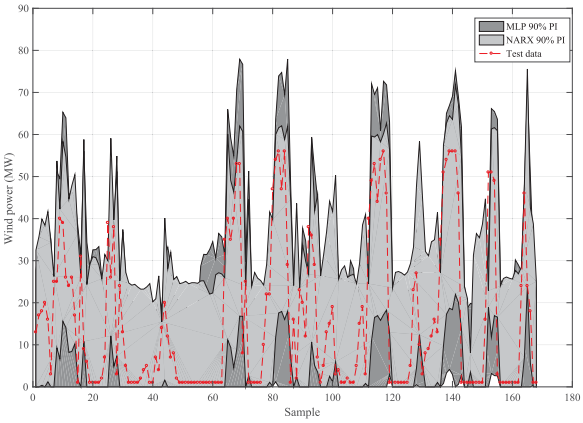

Fig. 6. Optimal PIs of NARX and MLP model.

TABLE III
PI CONSTRUCTION RESULTS OF NARX MODEL

No.	PICP(%)	PINAW(%)	PIMSE	NCWC	CWC(%)
1	94.03	65.23	1.5537	2.2060	65.23
2	94.12	68.78	1.6633	2.3511	68.78
3	94.20	69.72	1.5718	2.2690	69.72
4	93.47	63.23	1.3599	1.9922	63.23
5	93.04	68.63	1.6428	2.3290	**68.63**

this index for convenient comparison later. The median CWC value (63.71) instead of the best one is used to represent the average performance of Elman prediction model. Moreover, the median value is less influenced by outliers compared with the mean value.

As for NARX model, the PI construction is similar with that of Elman model. Hence, the results of only the last week are given in Fig. 6 for simplicity, from which we can obtain similar conclusions. The NARX model is also run for five times and the results are shown in Table III. As can be seen, the NARX model is also reliable and its average performance is represented by the median CWC value as well. The standard deviation of CWC for NARX model in Table III is 2.7613.

To further validate the effectiveness of the proposed method, we expand the dataset according to different seasons and test them with the proposed method, respectively. The average test results of four seasons with Elman model and NARX model are given in Table IV. From this table, we can see that both Elman model and NARX model can get good prediction results for different seasons. Moreover, the forecast results of spring and summer are better than those of autumn and winter by a comparison of CWC and NCWC values. This may be due to abrupt change of wind speed in autumn and winter seasons.

D. Comparison With Benchmark Models

For comparison purpose, some benchmark models are employed to construct PIs with the same dataset including naive method, ARIMA model, Gaussian process regression (GPR), QR and feedforward MLP model [22]. As a fundamental model, a typical three layer MLP was designed and the implementation procedure for PI construction is similar with that of RNN model. DA is also utilized to optimize the model with the same parameter settings. The optimal number of hidden nodes is determined to be 5 by five-fold cross-validation method. In addition, the activation functions are hyperbolic tangent and pure linear function for the hidden layer and output layer [22], respectively. The PI results of MLP model for the last week are shown in Fig. 6, where we can intuitively see that its interval width is larger than that of NARX model. To get quantitative results, MLP model is also repeated for five times and the median result is used to illustrate its average performance as shown in Table V. From Table V, it can be observed that the performance of MLP model seems not to be very stable due to the large standard deviation of CWC which is 3.9238.

ARIMA is a classical model used in time series forecast. Generally, ARIMA model performs better for one-step ahead forecast. In this study, the seasonal ARIMA model [47] is considered for direct one-step prediction [22]. The naive model is another well-known benchmark and it is similar to the persistence model in point forecast. According to this method, the forecast for the next step is generated based on the previous values. For example, the maximum and minimum values of the previous 20 samples are considered as the upper bound and lower bound for next step, respectively [33]. GPR model is an effective nonlinear prediction method which can be applied in many areas including regression and classification [48]. It adopts the Gaussian white noise assumption in the model and it is suitable to handle small sample problem. In addition, QR is another common statistical method that can be used for probabilistic forecast [49].

The comparison of PI construction results is summarized in Table VI. From Table VI, it can be observed that our proposed RNN prediction model outperforms the other benchmark models except for the GPR model. Note that in the GPR forecast model, the noise of the data is assumed to follow Gaussian distribution and the joint distribution of any finite number of variables is also Gaussian. However, the Gaussian distribution assumption is usually not applicable in practice. On the contrary, our method makes no assumptions on the data noise. In addition, the calculation time of GPR model for large dataset is very long and it is 3152.69 s in our case which is longer than the prediction time scale of one hour. Therefore, the application of GPR forecast model may not be feasible in practice.

In addition to the GPR model, Elman model achieves the lowest CWC value as well as the NCWC value, followed by the NARX model. All of the PICP values can satisfy the preassigned nominal coverage probability 90% except for the QR and naive model. Although the PINAW values are quite low for QR and naive model, their CWC values are very large due to the penalty on the unsatisfied PICP or low reliability. When PICP is satisfied, the PI quality depends on the interval width, which can be revealed by the comprehensive index CWC. By comparing the CWC values in Table VI, we can see that the PI quality has been significantly improved with the proposed RNN interval forecast model. The percentage improvements of Elman model are 14.02%, 21.74%, 62.71% and 84.90% in comparison with MLP, ARIMA, QR and naive model, respectively. They are

TABLE IV
PI CONSTRUCTION RESULTS FOR DIFFERENT SEASONS

Season	Elman model					NARX model				
	PICP(%)	PINAW(%)	PIMSE	NCWC	CWC(%)	PICP(%)	PINAW(%)	PIMSE	NCWC	CWC(%)
Spring	92.31	61.95	1.3643	1.9838	61.95	95.10	60.99	1.2494	1.8593	60.99
Summer	96.01	57.93	1.2364	1.8157	57.93	95.30	55.80	1.1230	1.6810	55.80
Autumn	91.73	66.30	1.4424	2.1054	66.30	91.25	65.68	1.8004	2.4572	65.68
Winter	94.93	64.50	1.3238	1.9687	64.50	95.17	67.02	1.3988	2.0690	67.02

TABLE V
PI CONSTRUCTION RESULTS OF MLP MODEL

No.	PICP(%)	PINAW(%)	PIMSE	NCWC	CWC(%)
1	95.92	70.65	2.2352	2.9416	70.65
2	96.13	74.10	1.6891	2.4302	**74.10**
3	96.99	76.97	1.7817	2.5514	76.97
4	96.56	72.36	1.6312	2.3547	72.36
5	95.53	80.57	2.4104	3.2161	80.57

TABLE VI
COMPARISON OF PROPOSED MODEL WITH BENCHMARK MODELS

Method	PICP(%)	PINAW(%)	PIMSE	NCWC	CWC(%)
Elman	93.81	63.71	1.4188	2.0559	63.71
NARX	93.04	68.63	1.6428	2.3290	68.63
MLP	96.13	74.10	1.6891	2.4302	74.10
ARIMA	92.01	81.41	2.0675	2.8816	81.41
GPR	90.51	43.12	0.6897	1.1209	43.12
QR	89.86	63.75	1.3373	3.0459	170.87
Naive	87.46	65.28	1.6303	5.8493	421.90

7.38%, 15.70%, 59.83% and 83.73%, respectively, for NARX model.

To verify the performance of DA optimization algorithm, we compare it with other population based approaches including GA and PSO. For the GA algorithm, the real-coded technique is employed and the probability of crossover and mutation are set to be 0.9 and 0.1, respectively [50]. The parameters of PSO algorithm are taken from [22]. With the same implementation procedure, we can obtain the average prediction results of Elman model and NARX model with GA and PSO algorithm which are summarized in Table VII. The training terminates when the maximum number of iterations is reached for all three algorithms, and the corresponding training times are also given in Table VII. As can be seen, the prediction results with the DA algorithm are the best according to the CWC and NCWC values. GA and PSO are easy to be trapped in local optima when the training terminates. In addition, the training time of DA and GA are almost the same, while the time of PSO is longer. The training time of the proposed method is about 10 minutes which is much less than the forecast time scale of one hour. In other words, the proposed method can be used for real-time forecast. Note that the execution time is related to the optimal structure of our model. If the number of hidden nodes in the recurrent model increases, then the average running time will also be longer.

E. Discussion

As the forecast time range prolongs, the forecast accuracy significantly decreases as a result of more uncertainties. Therefore, we only consider one-step ahead prediction in this work which is more accurate in practice. However, multi-step forecast may also be possible for our model with a proper data preprocess method. For wind power point forecast, we know that some other inputs such as NWP data and nearby wind speed are also considered except for the historical data to reduce the forecast error. These data may also be useful for interval prediction which is worth studying in the future. The delay embedding approach can be considered as a feature selection operation in our method. Also, other feature selection techniques can be used to process the input data which may help determine the most informative features and achieve higher quality PIs. Although it is possible to improve the forecast method as mentioned above, we can still find the superiority of our model from the comparison with other benchmark models.

Generally, the quality of PIs is evaluated by the coverage probability and interval width which are two conflicting objectives during PI construction process. To shorten the interval width may deteriorate the coverage probability and vice versa. Actually, the quality of PIs may be influenced by different factors such as data characteristics and forecast model. For different datasets, the PI results generated by the same forecast model are different as can be seen from Table IV. For forecast model, we may further study some modified RNN models or design a better comprehensive cost function in the existing single-objective framework to improve the PIs' quality in the future. In addition, considering the multiobjective characteristic of PI construction, a multiobjective problem formulation for interval prediction may also be a good choice for future research. In a single-objective framework for PI construction, the minimization of the comprehensive cost function NCWC or CWC is essentially to find a optimal compromise between these two aspects but it may not always balance them well. However, with a multiobjective framework, a set of Pareto optimal solutions can be generated and we can select a satisfactory one from them according to posterior preference information.

Due to the intrinsic randomness of DA algorithm in the training process, we may get different prediction results for each run. Therefore, to avoid a suboptimal solution, it is better to repeat the training several times and take the average as the optimal solution in a practical application. In addition, the PI results can be utilized in different ways in practice. For example, they can either be directly used in robust optimization problem of

TABLE VII
PI CONSTRUCTION RESULTS WITH DIFFERENT ALGORITHMS

Algorithm	Elman model						NARX model					
	PICP(%)	PINAW(%)	PIMSE	NCWC	CWC(%)	time(s)	PICP(%)	PINAW(%)	PIMSE	NCWC	CWC(%)	time(s)
DA	93.81	63.71	1.4188	2.0559	63.71	785.75	93.04	68.63	1.6428	2.3290	68.63	770.81
GA	94.85	74.78	1.8622	2.6100	74.78	771.49	94.07	71.48	1.7260	2.4408	71.48	770.80
PSO	93.21	86.30	2.5299	3.3930	86.30	1051.09	95.88	76.40	1.9980	2.7620	76.40	1022.86

power systems which only needs the lower and upper bound, or be transformed into deterministic point forecast values with convex combination of the lower and upper bound [51].

PI construction time is a critical factor in practice. Although the training time of the proposed RNN model is a little longer than MLP model, the test time, which is more useful for online applications, is almost the same with that of MLP model. The average PI construction time of Elman and NARX model in this study are 0.0770 s and 0.0737 s, respectively, which are very fast. All the experiments in this study are implemented with MATLAB software on a PC with hardware configuration of Intel Core TM i7-6700 CPU 3.40 GHz and 8 GB of RAM.

V. CONCLUSION

Renewable energy forecast is of great significance for system operation and scheduling in smart grid. Compared with point forecast, interval prediction is an efficient method to quantify the uncertainties in forecast. In this work, the RNN-based LUBE method is proposed to directly construct optimal PIs for wind power forecast. The RNN model has dynamic features and is suitable for time series forecast. Based on the single-objective problem formulation, a novel comprehensive cost function with a new PI evaluation index is designed to enhance the model training. To optimize the RNN prediction model, the DA algorithm with a new weight update method is introduced to solve the problem. In addition, the delay embedding theorem is applied to reconstruct the chaotic wind power data for better prediction. The numerical results with a real world wind power dataset show that the proposed RNN prediction model can construct better PIs compared with the benchmark models.

REFERENCES

[1] N. W. Miller, D. Guru, and K. Clark, "Wind generation," *IEEE Ind. Appl. Mag.*, vol. 15, no. 2, pp. 54–61, Mar./Apr. 2009.

[2] Canadian Wind Power Association (CANWEA). [Online]. Available: http://canwea.ca/wind-energy/national/. Accessed on May 15, 2017.

[3] H. B. Azad, S. Mekhilef, and V. G. Ganapathy, "Long-term wind speed forecasting and general pattern recognition using neural networks," *IEEE Trans. Sustain. Energy*, vol. 5, no. 2, pp. 546–553, Apr. 2014.

[4] E. B. Ssekulima, M. B. Anwar, A. Al Hinai, and M. S. El Moursi, "Wind speed and solar irradiance forecasting techniques for enhanced renewable energy integration with the grid: A review," *IET Renewable Power Gener.*, vol. 10, no. 7, pp. 885–989, 2016.

[5] P. Chen, T. Pedersen, B. Bak-Jensen, and Z. Chen, "Arima-based time series model of stochastic wind power generation," *IEEE Trans. Power Syst.*, vol. 25, no. 2, pp. 667–676, May 2010.

[6] H.-C. Wu and C.-N. Lu, "A data mining approach for spatial modeling in small area load forecast," *IEEE Trans. Power Syst.*, vol. 17, no. 2, pp. 516–521, May 2002.

[7] E. E. Elattar, "Prediction of wind power based on evolutionary optimised local general regression neural network," *IET Gener. Transm. Distrib.*, vol. 8, no. 5, pp. 916–923, 2014.

[8] S. Li, D. C. Wunsch, E. A. O'Hair, and M. G. Giesselmann, "Using neural networks to estimate wind turbine power generation," *IEEE Trans. Energy Convers.*, vol. 16, no. 3, pp. 276–282, Sep. 2001.

[9] J. Shi, W.-J. Lee, Y. Liu, Y. Yang, and P. Wang, "Forecasting power output of photovoltaic systems based on weather classification and support vector machines," *IEEE Trans. Ind. Appl.*, vol. 48, no. 3, pp. 1064–1069, May/Jun. 2012.

[10] N. Amjady, F. Keynia, and H. Zareipour, "Wind power prediction by a new forecast engine composed of modified hybrid neural network and enhanced particle swarm optimization," *IEEE Trans. Sustain. Energy*, vol. 2, no. 3, pp. 265–276, Jul. 2011.

[11] A. Khosravi and S. Nahavandi, "Combined nonparametric prediction intervals for wind power generation," *IEEE Trans. Sustain. Energy*, vol. 4, no. 4, pp. 849–856, Oct. 2013.

[12] P. Pinson, H. A. Nielsen, J. K. Møller, H. Madsen, and G. N. Kariniotakis, "Non-parametric probabilistic forecasts of wind power: Required properties and evaluation," *Wind Energy*, vol. 10, no. 6, pp. 497–516, 2007.

[13] Y. Zhang, J. Wang, and X. Wang, "Review on probabilistic forecasting of wind power generation," *Renew. Sustain. Energy Rev.*, vol. 32, pp. 255–270, Apr. 2014.

[14] J. B. Bremnes, "Probabilistic wind power forecasts using local quantile regression," *Wind Energy*, vol. 7, no. 1, pp. 47–54, 2004.

[15] P. Pinson and G. Kariniotakis, "Conditional prediction intervals of wind power generation," *IEEE Trans. Power Syst.*, vol. 25, no. 4, pp. 1845–1856, Nov. 2010.

[16] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong, "Optimal prediction intervals of wind power generation," *IEEE Trans. Power Syst.*, vol. 29, no. 3, pp. 1166–1174, May 2014.

[17] D. Sáez, F. Ávila, D. Olivares, C. Cañizares, and L. Marín, "Fuzzy prediction interval models for forecasting renewable resources and loads in microgrids," *IEEE Trans. Smart Grid*, vol. 6, no. 2, pp. 548–556, Mar. 2015.

[18] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Lower upper bound estimation method for construction of neural network-based prediction intervals," *IEEE Trans. Neural Netw.*, vol. 22, no. 3, pp. 337–346, Mar. 2011.

[19] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Comprehensive review of neural network-based prediction intervals and new advances," *IEEE Trans. Neural Netw.*, vol. 22, no. 9, pp. 1341–1356, Sep. 2011.

[20] A. Khosravi, S. Nahavandi, and D. Creighton, "Prediction interval construction and optimization for adaptive neurofuzzy inference systems," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 5, pp. 983–988, Oct. 2011.

[21] R. Ak, V. Vitelli, and E. Zio, "An interval-valued neural network approach for uncertainty quantification in short-term wind speed prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 11, pp. 2787–2800, Nov. 2015.

[22] H. Quan, D. Srinivasan, and A. Khosravi, "Short-term load and wind power forecasting using neural network-based prediction intervals," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 303–315, Feb. 2014.

[23] S. Qin, F. Liu, J. Wang, and Y. Song, "Interval forecasts of a novelty hybrid model for wind speeds," *Energy Rep.*, vol. 1, pp. 8–16, 2015.

[24] A. Kavousi-Fard, A. Khosravi, and S. Nahavandi, "A new fuzzy-based combined prediction interval for wind power forecasting," *IEEE Trans. Power Syst.*, vol. 31, no. 1, pp. 18–26, Jan. 2016.

[25] M. Khodayar and M. Teshnehlab, "Robust deep neural network for wind speed prediction," in *Proc. 4th Iranian Joint Congr. Fuzzy Intell. Syst.*, Sep. 2015, pp. 1–5.

[26] M. Khodayar, O. Kaynak, and M. E. Khodayar, "Rough deep neural architecture for short-term wind speed forecasting," *IEEE Trans. Ind. Informat.*, to be published, doi: 10.1109/TII.2017.2730846.
[27] S. Guo, Y. Li, and S. Xiao, "Wind speed forecasting of genetic neural model based on rough set theory," in *Proc. 5th Int. Conf. Crit. Infrastruct.*, Sep. 2010, pp. 1–6.
[28] C. M. Huang, Y. C. Huang, K. Y. Huang, S. J. Chen, and S. P. Yang, "Deterministic and probabilistic wind power forecasting using a hybrid method," in *Proc. IEEE Int. Conf. Ind. Technol.*, Mar. 2017, pp. 400–405.
[29] R. L. Welch, S. M. Ruffing, and G. K. Venayagamoorthy, "Comparison of feedforward and feedback neural network architectures for short term wind speed prediction," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2009, pp. 3335–3340.
[30] S. Anbazhagan and N. Kumarappan, "Day-ahead deregulated electricity market price forecasting using recurrent neural network," *IEEE Syst. J.*, vol. 7, no. 4, pp. 866–872, Dec. 2013.
[31] T. G. Barbounis, J. B. Theocharis, M. C. Alexiadis, and P. S. Dokopoulos, "Long-term wind speed and power forecasting using local recurrent neural network models," *IEEE Trans. Energy Convers.*, vol. 21, no. 1, pp. 273–284, Mar. 2006.
[32] J. Wang, W. Zhang, Y. Li, J. Wang, and Z. Dang, "Forecasting wind speed using empirical mode decomposition and Elman neural network," *Appl. Soft Comput.*, vol. 23, pp. 452–459, 2014.
[33] N. A. Shrivastava, A. Khosravi, and B. K. Panigrahi, "Prediction interval estimation of electricity prices using pso-tuned support vector machines," *IEEE Trans. Ind. Informat.*, vol. 11, no. 2, pp. 322–331, Apr. 2015.
[34] A. Khosravi, S. Nahavandi, and D. Creighton, "Prediction intervals for short-term wind farm power generation forecasts," *IEEE Trans. Sustain. Energy*, vol. 4, no. 3, pp. 602–610, Jul. 2013.
[35] J. L. Elman, "Finding structure in time," *Cognit. Sci.*, vol. 14, no. 2, pp. 179–211, 1990.
[36] M. Ardalani-Farsa and S. Zolfaghari, "Chaotic time series prediction with residual analysis method using hybrid Elman–NARX neural networks," *Neurocomputing*, vol. 73, no. 13, pp. 2540–2553, 2010.
[37] S. Mirjalili, "Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Comput. Appl.*, vol. 27, no. 4, pp. 1053–1073, 2016.
[38] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*. Beckington, U.K.: Luniver Press, 2010.
[39] R. C. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," in *Proc. IEEE Congr. Evol. Comput.*, 2001, vol. 1, pp. 94–100.
[40] Y. Wang, D. Wu, C. Guo, Q. Wu, W. Qian, and J. Yang, "Short-term wind speed prediction using support vector regression," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, 2010, pp. 1–6.
[41] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980*. New York, NY, USA: Springer, 1981, pp. 366–381.
[42] R. Hegger, H. Kantz, and T. Schreiber, "Practical implementation of nonlinear time series methods: The TISEAN package," *Chaos*, vol. 9, no. 2, pp. 413–435, Jun. 1999.
[43] Independent Electricity System Operator (IESO), Power Data. [Online]. Available: http://www.ieso.ca/en/power-data/data-directory. Accessed on Apr. 20, 2017.
[44] Canadian Wind Power Association (CANWEA). [Online]. Available: http://canwea.ca/wind-energy/ontario/. Accessed on May 18, 2017.
[45] Nonlinear Time Series Analysis (TISEAN). [Online]. Available: https://www.mpipks-dresden.mpg.de/tisean/. Accessed May 2017.
[46] Y. Hariya, T. Kurihara, T. Shindo, and K. Jin'no, "Lévy flight PSO," in *Proc. IEEE Congr. Evol. Comput.*, 2015, pp. 2678–2684.
[47] D. Yang, P. Jirutitijaroen, and W. M. Walsh, "Hourly solar irradiance time series forecasting using cloud cover index," *Sol. Energy*, vol. 86, no. 12, pp. 3531–3543, 2012.
[48] N. Chen, Z. Qian, I. T. Nabney, and X. Meng, "Wind power forecasts using Gaussian processes and numerical weather prediction," *IEEE Trans. Power Syst.*, vol. 29, no. 2, pp. 656–665, Mar. 2014.
[49] R. Koenker and K. Hallock, "Quantile regression: An introduction," *J. Econ. Perspectives*, vol. 15, no. 4, pp. 43–56, 2001.
[50] A. Blanco, M. Delgado, and M. C. Pegalajar, "A real-coded genetic algorithm for training recurrent neural networks," *Neural Netw.*, vol. 14, no. 1, pp. 93–105, 2001.
[51] F. Valencia, J. Collado, D. Sáez, and L. G. Marín, "Robust energy management system for a microgrid based on a fuzzy prediction interval model," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1486–1494, May 2016.

Zhichao Shi (S'17) received the B.Sc. and M.Sc. degrees from Nankai University, Tianjin, China, and the National University of Defense Technology, Changsha, China, in 2013 and 2015, respectively. He is currently working toward the Ph.D. degree in electrical and computer engineering at the University of Alberta, Edmonton, AB, Canada. His research interests include smart grid, computational intelligence and optimization, and renewable energy system.

Hao Liang (S'09–M'14) received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in 2013. Since 2014, he has been an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Alberta, Edmonton, AB, Canada. His research interests include smart grid, wireless communications, and wireless networking.

Venkata Dinavahi (SM'08) received the Ph.D. degree from the University of Toronto, Toronto, ON, Canada, in 2000. He is currently a Professor in the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. His research interests include real-time simulation of power systems and power electronics systems, large-scale system simulation, and parallel and distributed computing.