Deep Concatenated Residual Network With Bidirectional LSTM for One-Hour-Ahead Wind Power Forecasting

Min-Seung Ko, *Student Member, IEEE*, Kwangsuk Lee, *Member, IEEE*, Jae-Kyeong Kim, *Member, IEEE*, Chang Woo Hong, *Graduate Student Member, IEEE*, Zhao Yang Dong, *Fellow, IEEE*, and Kyeon Hur, *Senior Member, IEEE*

***Abstract*—This paper presents a deep residual network for improving time-series forecasting models, indispensable to reliable and economical power grid operations, especially with high shares of renewable energy sources. Motivated by the potential performance degradation due to the overfitting of the prevailing stacked bidirectional long short-term memory (Bi-LSTM) layers associated with its linear stacking, we propose a concatenated residual learning by connecting the multi-level residual network (MRN) and DenseNet. This method further integrates long and short Bi-LSTM networks, ReLU, and SeLU for its activating function. Rigorous studies present superior prediction accuracy and parameter efficiency for the widely used temperature dataset as well as the actual wind power dataset. The peak value forecasting and generalization capability, along with the credible confidence range, demonstrate that the proposed model offers essential features of a time-series forecasting, enabling a general forecasting framework in grid operations. The source code of this paper can be found in https://github.com/MinseungKo/DRNet.git.**

***Index Terms*—Activation function, bidirectional learning, deep learning, long short-term memory, residual networks, wind power forecasting.**

I. Introduction

THE increasing concerns about sustainable environment and energy systems have led to the widespread growth of renewable energy resources such as solar and wind energy, and significant change in the composition of the electricity generation mix [1]–[4]. The highest annual growth of these resources can be observed all over the world and manifests the fast energy transition [5]. For example, the worldwide wind power capacity has grown from 180 GW in 2010 to 622 GW in 2019 [6]. The solar power capacity has concurrently grown from 41 GW to 585 GW [6]. Solar and wind penetrations are expected to grow further, owing to their improved economic benefits. This paper thus uses "variable renewable energy (VRE)" to represent solar and wind energy [7], [8].

The weather-dependent variability of these energy resources [9], however, may threaten the reliability and economical efficiency of power system operations, leading to significant social and economic losses [3], [10], [11]. Among the various methods to handle the supply-side variability, VRE forecasting is the most fundamental and practical front-end application. Its accuracy facilitates a secure and economical grid integration of the VRE [7], [10]. Compared to solar, it has been understood that wind power is less predictable because of its highly uncertain characteristics [7]. Besides, wind generators tend to be installed as wind power plants rather than distributed generators, unlike the solar generators [12], [13]. This geographical aggregation and smoothing help reduce operating reserve requirements, particularly beneficial to the bulk power system operations. Various studies have thus been conducted to investigate the power system impact of the aggregated wind power and methods for improving the wind power forecasting (WPF) [14]–[16].

WPF methods can be classified into three categories: physical method, conventional statistical method, and artificial neural network (ANN) based method. A hybrid one combining more than two methods above has been investigated to complement each other [3], [17]–[19]. The physical method builds upon the meso-scale weather model or the numerical weather prediction system (NWP), which represents the mathematically expressive model based on various geographical and meteorological information [20], [21]. Though this method performs good for medium-term forecasting periods of more than 3 hours, it has limitations on short-term forecasting because of the difficulty in gathering all the related geographical or meteorological data [21]–[23].

Manuscript received May 13, 2020; revised September 3, 2020 and November 4, 2020; accepted December 2, 2020. Date of publication December 10, 2020; date of current version March 22, 2021. This work was supported in part by the Framework of International Cooperation Program Managed by National Research Foundation of Korea under Grant 2017K1A4A3013579 and in part by the "Human Resources Program in Energy Technology" of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) Granted Financial Resource from the Ministry of Trade, Industry & Energy, Republic of Korea under Grant 20194030202420. Paper no. TSTE-00500-2020. *(Corresponding author: Kyeon Hur.)*

Min-Seung Ko, Chang Woo Hong, and Kyeon Hur are with the School of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, Korea (e-mail: kms4634500@yonsei.ac.kr; spearw@yonsei.ac.kr; khur@yonsei.ac.kr).

Kwangsuk Lee is with SK Telecom, Seoul 04539, Korea (e-mail: kwangsuklee@yonsei.ac.kr).

Jae-Kyeong Kim is with the Advanced Power Grid Research Center, Korea Electrotechnology Research Institute (KERI), Uiwang 16029, Korea (e-mail: jkkim@keri.re.kr).

Zhao Yang Dong is with the School of Electrical Engineering and Telecommunications, University of New South Wales, Kensington, NSW 2052, Australia (e-mail: joe.dong@unsw.edu.au).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSTE.2020.3043884.

Digital Object Identifier 10.1109/TSTE.2020.3043884

The conventional statistical method produces a linear characteristic of the wind power output based on the historical data [24]. Well-known methods such as AR or ARIMA models have been widely used to construct a linear relationship; however, the nonlinearity of the data often compromises the accuracy and generality of the model [17]. Though there are various approaches to express the nonlinearity based on the conventional statistical methods, these methods still based on the linear forms are limited in representing the nonlinear dynamics [25]–[27]. On the other hand, the ANN-based method can effectively represent the nonlinear and complex features of wind speed and power with a large number of parameters.

ANN-based forecast methods have been widely used with the improvement of memories and arithmetic units. ANN-based shallow models for WPF or wind speed forecasting (WSF) are proposed with higher accuracy than physical or conventional statistical methods [28], [29]. Hybrid models, paralleling basic ANN models with other models, e.g., Kalman filters, and support vector machines, are proposed to boost the accuracy of WSF [30], [31]. The introduction of a recurrent neural network (RNN) in deep neural network (DNN) improved the accuracy of ANN models [32], [33]. Furthermore, long short-term memory (LSTM) network, which is the advanced structure of RNN, is introduced. The memory cell of LSTM helps significantly decrease the time-series forecasting error [34]. Based on the ability to keep the data for a long time, LSTM is used to extract temporal features for WSF [35]. The LSTM based WSF models outperform the ANN or ARIMA based models as demonstrated in [17], [36]. However, these previous studies mainly focused on obtaining the diverse information from various LSTM networks with few LSTM cells and the benefits from the deep learning were not fully exploited.

In general, the performance of DNN increases as the network depth grows. However, after a specific size of the network, overfitting issues can arise and negatively affect the overall DNN performance [37], [38]. Two approaches have been taken to mitigate these problems: ameliorating the layer itself and transforming the structure of DNN [38]–[42]. A representative method of the first approach is bidirectional LSTM (Bi-LSTM) [39]. Unlike LSTM training only in a forward direction, Bi-LSTM allows for bidirectional training to improve the performance of sequence learning [40]. In [43], [44], Bi-LSTM networks for WPF and WSF achieve higher forecasting accuracy than LSTM networks. However, it should be noticed that Bi-LSTM networks do not always outperform LSTM networks, which is handled in Section IV of this paper. As an example of the second approach, residual learning modifies the structure of DNN with shortcut connections and effectively trains DNN. Variants of residual learning have been proposed and reported improved performance [38], [41], [42]. However, the second approach is only limited to CNN for sequential data. Structural improvement of RNN is desired for highly complex sequential data, as the network needs to be deeper to handle the data.

This paper thus proposes deep concatenated residual networks (DRNets) for RNN as detailed in Section III following the brief discussion of DNN, RNN, and (Bi-)LSTM in Section II. DRNets integrate the key concepts of DenseNet and multi-level residual network and further incorporate several new improvements, including activation functions and fused structure with short and long Bi-LSTMs. The adequate constitution of RNN layers is firstly investigated when DRNets are employed. With the constitution, the combination of ReLU and SeLU is proposed for activating the network. Finally, the fused concept, which exploits results from both short and long Bi-LSTMs, is used to enhance the peak value forecasting capability. In particular, the proposed model is adopted for 1-h ahead aggregated WPF, useful for a range of system operations, including scheduling, dispatch, and operating reserve requirements [7]. The accuracy and efficacy of the proposed forecasting model are demonstrated through rigorous case studies using the historical wind power data from ERCOT[1] and validated by the other type of Jena's temperature data, as presented in Section IV. Finally, concluding remarks are provided in Section V.

II. Bi-LSTM Forecasting Network

A. Deep Neural Network (DNN) and Recurrent Neural Network (RNN)

The DNN is an improved model of ANN with multiple processing layers to learn representations of data [46]. Technological improvement in memories and arithmetic units enables DNN to express high dimensional data. DNN can be understood as a large black box function, which even trains the functional form itself. The DNN thus embraces the complex nonlinear dynamics of the wind power, without providing the functional structure for each dynamic pattern. In addition, various uncertain weather factors commonly affect the wind power plants and their power outputs to a varying degree, which cannot be adequately captured by the shallow networks [47]. Therefore, compared to ANN, DNN should be more adequate for WPF with the following two attributes: 1) Ability to learn common or shared uncertainties and 2) Ability to learn nonlinear relationships [47].

The RNN is one category of DNN suitable for the sequence learning. RNN receives a sequence as an input at a time and maps the sequence with the sequential output [39], as featured in the recent successful applications such as speech recognition, natural language translation, and image captioning [48], [49]. The conventional structure of RNN with N layers and its unfolded graph are shown in Fig. 1. For data input x_t at time step t, corresponding predicted output O_t can be represented as the following equations:

$$h_l^t = g_l^t(U_t \cdot x_t + W_l \cdot h_l^{t-1} + b_x) \tag{1}$$

$$O_t = g_N^t(V_N^t \cdot h_N^t + b_y) \tag{2}$$

where g_l^t represents the activation functions of the lth layer at t, and $l = 1, 2, \ldots, N$. b_x and b_y are bias terms, U_t, W_l, and V_N are weight matrices, and h_l^t is the sharing state vector of lth layer. For each time step, parameters of RNN are updated to minimize the value of the loss function $L(O_t, y_t)$, where y_t is the desired

[1] ERCOT also conducts WPF with the time step of 5 min for real-time dispatch [45] This paper focuses on 1-h ahead WPF, particularly useful to cope with the larger variation.

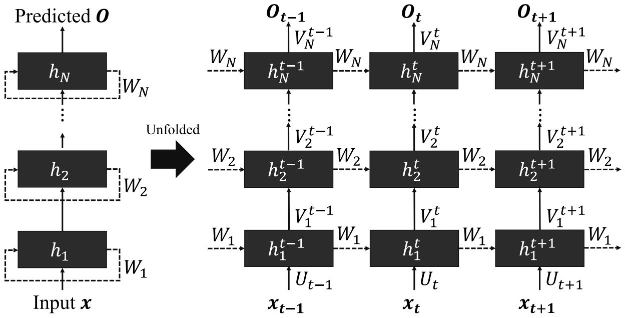


Fig. 1. Conventional structure and unfolded graph of RNN with N layers.

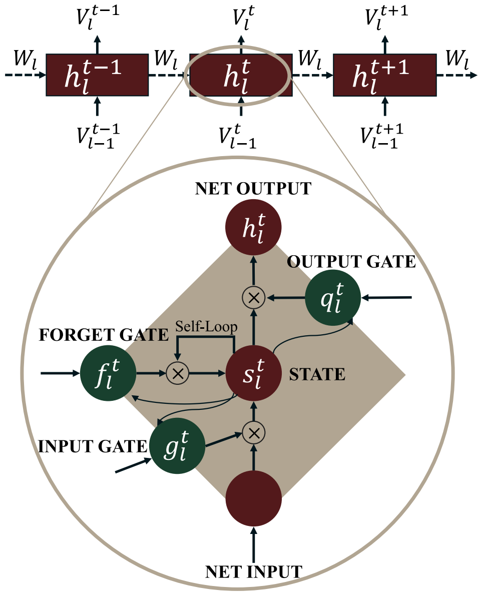


Fig. 2. General structure of LSTM cell.

output. Therefore, RNN can learn temporal features owing to its sharing property coming from the state vector.

B. Long Short-Term Memory (LSTM) and Bidirectional LSTM (Bi-LSTM)

The LSTM is an improved architecture of conventional RNN to overcome its limitation on solving problems with long-term dependency [50]. LSTM can alleviate the vanishing gradient problem by adding a special hidden unit, known as a memory unit. This unit can accumulate or remove the new inputs. Three controlling gates determine the operation of the unit by controlling the flow of data, as illustrated in Fig. 2 [51]. The forget gate discards useless memories from the state vector and the input gate adds the necessary information from the new input and previous net output. Finally, the new output of the corresponding unit is determined by the output gate. Based on these operations, LSTM unit can keep the useful data for a long duration, so it captures the long-term dependencies better than the conventional RNN.

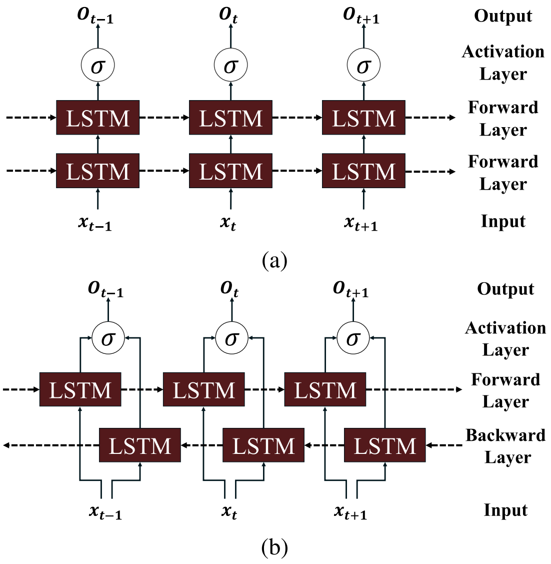


Fig. 3. General structure of (a) deep LSTM network and (b) Bi-LSTM network.

Bidirectional learning can contribute to boosting the accuracy of conventional RNN [52]. Bidirectional learning has been adopted from the reasoning that the output is not a sole product of previous inputs, but a piece of the continuous correlation. Bidirectional RNN (BRNN) trains its parameters in both forward and reverse paths to understand the context. This training process can capture the features or patterns in bidirectional aspects, whereas RNN is trained in the forward-path only. BRNN showed higher accuracy and performance in sequence learning than conventional RNN, especially in speech processing tasks [52].

As represented in Fig. 3, Bi-LSTM incorporates the bidirectional concept into LSTM. The forward layer of Bi-LSTM updates the parameters in the unit as does the LSTM network. On the other hand, LSTM cells in the backward layer compute the derivative of the propagated errors in the forward layer. The operation of a single LSTM cell h in a backward layer can be described as follows [39], [51]:

$$\epsilon_h^t = \sum_{i \in N} U_h^i \cdot \delta_i^{t+1} \tag{3}$$

$$\delta_q^t = {y_q^t}' \cdot \sum_{h \in C} (\epsilon_h^t \cdot \tanh(s_h^t)) \tag{4}$$

$$\begin{aligned}\frac{\partial E^t}{\partial s_h^t} &= \epsilon_h^t \cdot y_q^t \cdot \frac{\partial \tanh(y_h^t)}{\partial y_h^t} + \frac{\partial E^{t+1}}{\partial s_h^{t+1}} \cdot y_f^{t+1} \\ &\quad + \delta_g^{t+1} \cdot W_h^g + \delta_f^{t+1} \cdot W_h^f + \delta_q^t \cdot W_h^q \end{aligned} \tag{5}$$

$$\delta_h^t = y_g^t \cdot {y_h^t}' \cdot \frac{\partial E^t}{\partial s_h^t}$$

$$\delta_f^t = {y_f^t}' \cdot \sum_{h \in C} \left(\frac{\partial E^t}{\partial s_h^t} \cdot y_q^{t+1} \cdot \tanh\left(s_h^{t+1}\right) \right) \tag{6}$$

$$\delta_g^t = {y_g^t}' \cdot \sum_{h \in C} \left(\frac{\partial E^t}{\partial s_h^t} \cdot y_h^t \right) \tag{7}$$

where ϵ_h^t is the backpropagated error of the output on cell h at time t, N is the set of all units, U and W denote weight matrices, and C is the set of cells. δ is the derivative of error regarding the gates of the cell, and y is the output of each gate, where subscripts g, s, f, and q represent input gate, state, forget gate, and output gate each. Especially, cell output y_c is determined by $\tanh$ function, while the other outputs are calculated using a sigmoid function. s_h is the state value of h, and E^t is the net output error at time t. According to the operation of LSTM unit in the backward layer, Bi-LSTM can adjust the parameters to lessen the propagated errors in the forward layer.

III. Proposed Residual Learning

A. Conventional Residual Learning

Though stacking layers enables DNN to enrich the level of features, other problems may arise [53]. At the early stage of DNN, a vanishing/exploding gradient problem may adversely affect the convergence of DNN during the training process. Degradation and overfitting are the newly exposed problems after resolving the convergence issues. The radical root of these problems is the network not adequately structured to train its parameters. Degradation is the retrogression of training related to the network depth. As the network depth increases, training error of the network saturates and then increases rapidly. On the other hand, overfitting is related to validation or test. Overfitting occurs as the parameters are updated for the training dataset only; the performance of the trained network on test dataset thus decreases.

The most fundamental way to avoid these problems is to secure more training data or to reduce the size of networks. However, there are some limitations on boosting the size of training data in reality, and reducing the size of networks can degrade the adaptability of networks to the other types of data. Therefore, various methods, such as dropout strategy and pooling strategy, have been proposed to solve the problems. Dropout, one of the most successful regularization methods, reorganizes the network only with strongly related connections by evaluating each unit with random masking during the training process [54]. In addition, pooling strategy can reduce the network features by mapping more than 2D to a single output [55]. However, the noise added on RNN layer using the dropout can be amplified with the depth of the network. It is thus advised that the dropout should be applied only for shallow RNN [56], [57]. Pooling strategy is not widely adopted for RNN, because RNN is the sequence pooling process in itself [58], [59].

Among the countermeasures, residual learning has been widely used due to its simplicity and effectiveness [60]. The basic idea of the residual learning connects shortcuts within the network. The desired mapping of the stacked layers in conventional DNN can be represented as

$$\mathcal{H}(x) = \mathcal{F}(x) \tag{8}$$

where $\mathcal{H}(x)$ is the desired mapping of stacked layers, $\mathcal{F}$ denotes the pure stacked layers, and x is the input of the stacked layers. In case of the residual, the representation of mapping is different from one of the conventional DNN as

$$\mathcal{H}(x) = \mathcal{F}(x) + x. \tag{9}$$

The addition of x in (9) represents that the residual learning can be regarded as a feedforward network. When the whole network is composed of n stacked layers, conventional DNN is just an expression between x and $\mathcal{H}_n$. Therefore, the total network aims to lessen $\mathcal{H}_n - x$. On the other hand, if the residual learning is applied to each stacked layer, ith stacked layer optimizes its weight to drive $\mathcal{H}_i - \mathcal{H}_{i-1}$ to zero, where $i = 1, 2, \ldots, n$. Therefore, the network with the residual learning is easier to optimize weights compared to the conventional network, because the residual network can be regarded to divide the optimization tasks. In addition, the final output of the residual network can be represented as $\mathcal{F}_n(\mathcal{H}_{n-1}) + \cdots + \mathcal{F}_1(x) + x$, while the output of the conventional network is $\mathcal{F}_n(\mathcal{F}_{n-1}(\cdots \mathcal{F}_1(x)))$. This shows that the residual network can maintain the input flow, not stuck in the training details.

Accuracy can further be improved by modifying the residual learning with various strategies [42], [61]–[63]. Multi-residual network incorporates additional shortcuts on ResNet. The identity mapping of the network can be multiple levels [61]. Another approach to reform the residual network is the fused network. Fused network stacks the layers in both vertical and horizontal directions, which resembles the ensemble approach [62]. Multilevel residual network (MRN) and DenseNet are the improved residual learnings widely used in CNN. Instead of identity mapping, MRN uses 1D CNN mapping based on the hypothesis that the residual mapping of the residual network can be optimized [63]. DenseNet connects all the layers by concatenation, unlike the other structures whose layers are connected through addition [42]. However, adding layers of MRN is limited in improving the WPF performance, compared to concatenation. Unlike MRN, the size of the network increases geometrically with the increase of layer as DenseNet concatenates all the layers.

B. Proposed Residual Learning

To overcome the limitations, this paper proposes another structure of residual learning, which combines the strengths of MRN and DenseNet. The key concepts of the proposed residual networks (DRNets) are shown in Fig. 4. For MRN with stacked layers and total 2n layers, the output can be represented as follows:

$$\mathcal{H}_n = \mathcal{F}_n(\mathcal{H}_{n-1}) + Act(J(\mathcal{H}_{n-1})) + Act(J(x)) \tag{10}$$

where $J(k)$ represents the output of 1D CNN for input k, and $Act(x)$ denotes an activated output for input x. In case of DenseNet, the output of the ith stacked layer becomes

$$\mathcal{H}_i = \mathcal{F}_i(\mathcal{H}_{i-1}) \parallel \mathcal{H}_{i-1} \parallel \mathcal{H}_{i-2} \parallel \cdots \parallel x \tag{11}$$

where $\parallel$ means the concatenation. Therefore, the output of the ith stacked layer with DRNets can be expressed as the combination

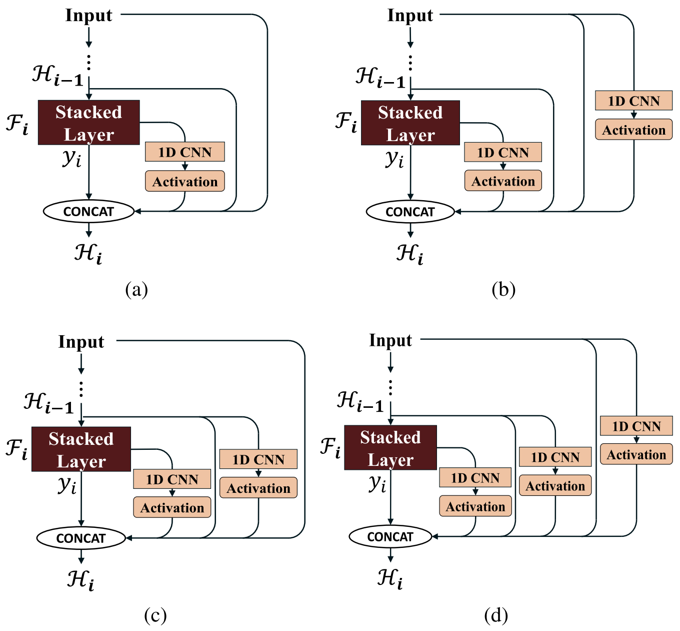


Fig. 4. Key concepts of (a) DRNet-1, (b) DRNet-2, (c) DRNet-3, and (d) DRNet-4.

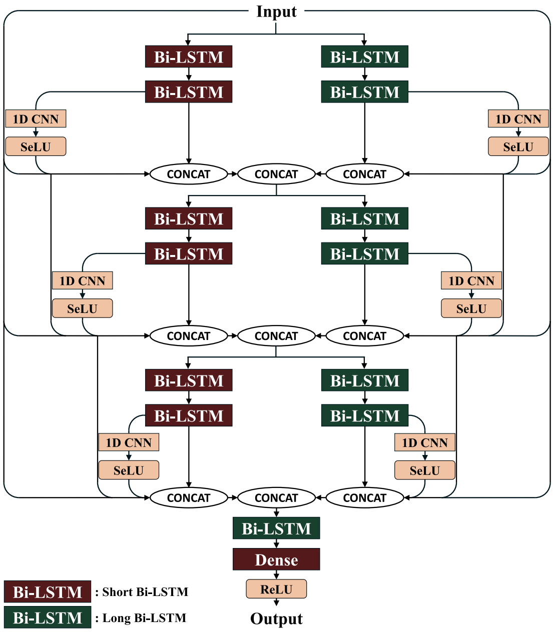


Fig. 5. Structure of the proposed forecasting model using DRNet-1 and Bi-LSTM layers.

of (10) and (11) as follows:

$$\begin{aligned}
\mathcal{H}_i &= y_i \parallel Act(J(y_i)) \parallel \mathcal{H}_{i-1} \parallel x : \text{DRNet-1} \\
\mathcal{H}_i &= y_i \parallel Act(J(y_i)) \parallel \mathcal{H}_{i-1} \parallel x \\
&\qquad\qquad \parallel Act(J(x)) : \text{DRNet-2} \\
\mathcal{H}_i &= y_i \parallel Act(J(y_i)) \parallel \mathcal{H}_{i-1} \parallel x \qquad (12) \\
&\qquad\qquad \parallel Act(J(\mathcal{H}_{i-1})) : \text{DRNet-3} \\
\mathcal{H}_i &= y_i \parallel Act(J(y_i)) \parallel \mathcal{H}_{i-1} \parallel x \\
&\quad \parallel Act(J(x)) \parallel Act(J(\mathcal{H}_{i-1})) : \text{DRNet-4}
\end{aligned}$$

$$y_i = \mathcal{F}_i(\mathcal{H}_{i-1}) \qquad (13)$$

where y_i is the pure output of ith stacked layer.

Structural formulations in (12) present that DRNets use 1D CNN mappings and the activation functions similar to MRN, but do not connect all the outputs from residuals. At the same time, the shortcuts gather through concatenation like DenseNet. Therefore, DRNets can nurture both the effective activation of MRN and the preservation of data of DenseNet. In addition, the total number of parameters is maintained similar to those of MRN and DenseNet because the numerical increase of parameters by concatenation is covered by fewer links of the residual. Another feature of DRNets differentiated from the others is that the inputs of the concatenation include not only y_i but also activated y_i, which is expressed as $Act(J(y_i))$. In turn, DRNets perform well with higher parameter efficiency than the other residual learnings.

C. Additional Improvements: Peak Value Forecasting and Confidence Interval

Peak load forecasting has been regarded as an essential tool to make decisions related to the power system operations [64]–[66]. Because the wind generation can be considered as negative load in the steady-state operations, predicting the peak value of wind power outputs should particularly be beneficial in estimating standby capacity of the power system and load rates with capacity factors, i.e., useful information for unit commitment or deploying quick start generators. With the increase of wind power penetration level, the importance of the accurate peak value forecasting should increase.

To further improve peak value forecasting capability, the proposed model adopts the fused concept, as illustrated in Fig. 5. The horizontally stretched size of RNN layers is related to the period during which the layers can learn best. Horizontally long RNN has strength in apprehending long-term tendencies while short RNN learns short-term tendencies well. The fused net, composed of long and short Bi-LSTM networks, can nurture both strengths. A single Bi-LSTM layer at the end of the network determines the participation of long and short Bi-LSTM layers, which is related to the impact of short and long-term uncertainties on the output. Therefore, the fused concept helps the model better analyze the sequence and improve the peak value forecasting of wind power.

Types of activation functions influence the forecasting performance. One of the most widely used activation functions is ReLU, which has significantly improved the performance of deep neural networks [67]. However, ReLU has a serious problem known as a dying ReLU problem. When ReLU activates

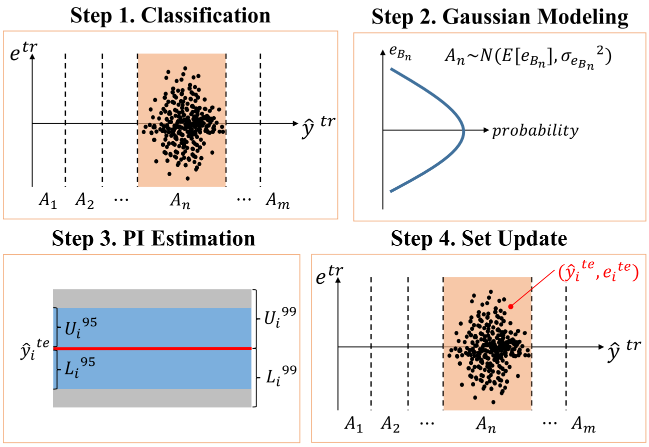


Fig. 6. Overall process of computing probabilistic intervals.

a large portion of hidden units as 0, the gradient-based algorithms cannot update the weights. In order to solve this problem, leaky ReLU and eLU were proposed [68], [69]. The SeLU or scaled exponential linear unit has one more tunable parameter than eLU [69]. On top of a tunable parameter α of eLU, SeLU has another tunable parameter λ, which can be represented as

$$\mathrm{SeLU}(m) = \lambda \begin{cases} m & \text{if } m > 0, \\ \alpha e^m - \alpha & \text{if } m \leq 0 \end{cases}. \tag{14}$$

The SeLU not only avoids the dying ReLU problem but also offers a self-normalizing characteristic because the activation of a normally distributed input through SeLU converges towards the normal distribution [69], [70]. Therefore, SeLU helps train deep networks without gradient problems. Activation functions of the proposed model are divided into two categories. The first one is the function used for activating 1D CNN layer, and the other is used for activating Dense layer. As shown in Fig. 5, the proposed activation functions (called as the final ReLU), take SeLU for the former category and ReLU for the latter.

Because there always exist forecasting errors and uncertainties, quantifying the confidence interval about the predicted value should be helpful for more reliable operation of the power system [71], [72]. As represented in Fig. 6, the overall process of obtaining stochastic or probabilistic intervals (PIs) consists of 4 steps; Classification, Gaussian Modeling, PI Estimation, and Set Update. From the training results, elements specified by the predicted values in a training set, $\hat{y}^{tr}$, and the prediction error $e^{tr} = y^{tr} - \hat{y}^{tr}$ can be obtained. If $\hat{y}^{tr}$ is matched to x-axis, and e^{tr} to y-axis, each element can be expressed as a form of ($\hat{y}^{tr}$, e^{tr}). According to the value of $\hat{y}^{tr}$, the elements can be classified into m sets. The value m should be selected so that each set A_k has sufficient elements for assuming Gaussian distribution. If we let B_k represent the elements of A_k, e_{B_k} are the prediction errors of B_k, and $k \in [1, 2, \ldots, m]$, A_k can be assumed to follow Gaussian distribution with $E[e_{B_k}]$ and variance $\sigma_{e_{B_k}}{}^2$, i.e., $A_k \sim N(E[e_{B_k}], \sigma_{e_{B_k}}{}^2)$. For ith predicted value in the test set, the prediction error $e_i{}^{te}$ can be estimated by the corresponding Gaussian distribution of the training set. Assuming that $\hat{y}_i^{te}$ lies in the x-axis range of A_n, an stochastic interval $I_{e_i}^{te}$ about the forecasting error with $100(1-\beta)\%$ confidence level can be expressed as

$$I_{e_i}^{te} = [L_i{}^\beta(A_{n,i}), U_i{}^\beta(A_{n,i})] \tag{15}$$

where the lower bound $L_i{}^\beta$ and the upper bound $U_i{}^\beta$ with standard score $z_{1-\beta/2}$ can be calculated as

$$L_i{}^\beta = E[e_{B_{n,i}}] - z_{1-\beta/2}\sqrt{\sigma_{e_{B_{n,i}}}{}^2} \tag{16}$$

$$U_i{}^\beta = E[e_{B_{n,i}}] + z_{1-\beta/2}\sqrt{\sigma_{e_{B_{n,i}}}{}^2}. \tag{17}$$

Finally, the confidence interval about the wind power forecasting can be obtained as below:

$$I_i^{te} = [\hat{y}_i^{te} + L_i{}^\beta, \hat{y}_i^{te} + U_i{}^\beta]. \tag{18}$$

The above equations denote that the confidence interval for specific test prediction data is determined by the interval of corresponding set, which is initially composed of training prediction data and error. After the real value of the test set, $y_i{}^{te}$, is revealed, the next prediction error set $A_{n,i+1}$ is updated with $e_i^{te} = y_i{}^{te} - \hat{y}_i{}^{te}$. Therefore, the corresponding set $A_{n,i+1} = [A_{n,i}, (\hat{y}_i{}^{te}, e_i^{te})]$, and the other sets $A_{k,i+1} = A_{k,i}$ for $k \in [1, 2, \ldots, m] \cap [n]^C$. Then the new test confidence interval is determined based on the new test prediction value, $y_{i+1}{}^{te}$, and the updated m sets.

TABLE I

Summary of the Test Settings

Training method	Adam	Batch size	40
Loss function	MSE, MAE, MAPE	Input sequence length	{24}
Training strategy		ReduceLROnPlateau	

IV. Case Study

A. Test Settings

This paper presents 1-h ahead forecasting on the wind power dataset of ERCOT and Jena's temperature dataset. Each dataset is composed of hourly average data with a single feature without any other variables. The values of each dataset are preprocessed to fit in the range between 0 and 1, for example, by MinMaxScaler in Python [73]. Each dataset is then divided into training, validation, and test sets. The training set updates the parameters of the forecasting models, and the validation set selects the best-performed model. The test set evaluates the performance of the selected model. In order to optimize the model during the training process, three widely used metrics, including MSE, MAE, and MAPE, are employed. Note that all the metrics are calculated based on the preprocessed data. Test settings are summarized in Table I.

The overall timeline of the proposed WPF is illustrated in Fig. 7. In this paper, update time and forecasting horizon are set to be 1-h to assist, e.g., the hourly reliability unit commitment. For k-h WPF, all the corresponding data would be collected at $(k-1)$-h plus data acquisition time. At the same time, k-h wind power output forecasting is performed; thus, the actual lead time

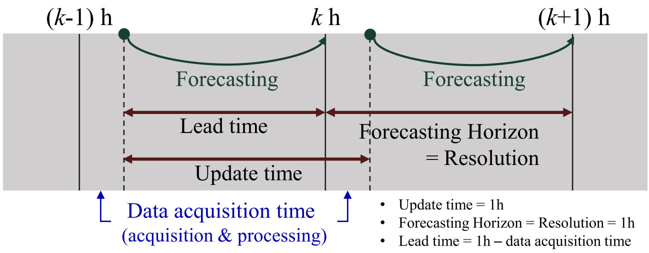

(k-1) h
k h
(k+1) h
Forecasting
Forecasting
Lead time
Forecasting Horizon = Resolution
Update time
Data acquisition time
(acquisition & processing)
Update time = 1h
Forecasting Horizon = Resolution = 1h
Lead time = 1h – data acquisition time

Fig. 7. Timeline for the proposed wind power forecasting.

TABLE II
PERFORMANCE OF THE PROPOSED NETWORK ON JENA'S TEMPERATURE FORECASTING

Depth	Residual Learning	Training Error	Validation Error		
		MSE ($\times10^{-4}$)	MSE ($\times10^{-4}$)	MAE ($\times10^{-3}$)	MAPE (%)
11	Pure Bi-LSTM	9.1526	8.9575	2.2670	4.308
	DenseNet	2.4632	2.4221	1.1371	2.2106
	DRNet-3	2.0064	1.9879	1.0047	1.9118
	Fused DRNet-1	**1.8318**	**1.7953**	**0.9511**	**1.8277**
Depth	Residual Learning	Test Error			Mean of the largest 10% Error
		MSE ($\times10^{-4}$)	MAE ($\times10^{-3}$)	MAPE (%)	
11	Pure Bi-LSTM	8.3107	2.1950	3.9463	0.4680
	DenseNet	2.2763	1.0978	2.0041	0.2244
	DRNet-3	1.8942	0.9863	1.7673	0.2313
	Fused DRNet-1	**1.7143**	**0.9310**	**1.6750**	**0.1586**

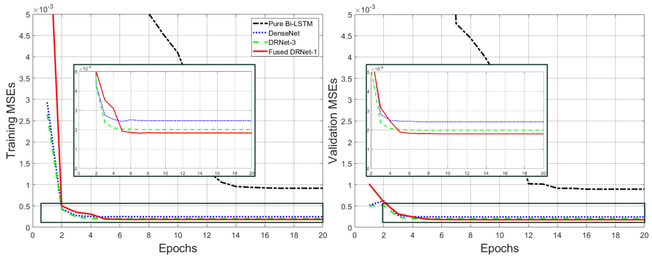


Fig. 8. Training curves of various networks for temperature forecasting.

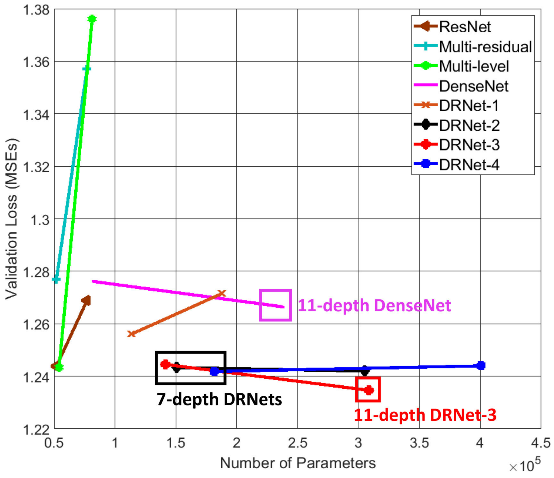


Fig. 9. Parameter efficiency of various residual learnings.

should be 1-h minus data acquisition time. The forecasting resolution is the same as the forecasting horizon based on the hourly dataset. Higher resolution or intra-hour forecasting could be achieved with the shorter update time. The forecasting horizon could also be extended for various applications in operational planning.

B. Temperature Forecasting Results

To secure the objectivity of the proposed model, we have conducted forecasting experiments on the temperature dataset of Jena in Germany, which has been widely used for RNN performance test [74]. The temperature data embeds variability and imposes forecasting uncertainty, similar to the wind power data. The experiments are based on the data of 70037 hours from 2009 to 2017 with 60%, 30%, and 10% data division for training, validation and test set. Table II represents the experimental results of the 1-h ahead temperature forecasting. DRNet-3 shows better performance than DenseNet in all metrics except the peak value forecasting. Fused DRNet-1 improves both overall performance and the peak value forecasting more than DRNet-3. Moreover, the standard deviation of errors with fused DRNet-1 is 1.1957, which is smaller than 1.2085 and 1.3447 with DRNet-3 and DenseNet, respectively. These results imply both the improved performance and the generalization capability of the fused DRNet for a time-series.

Training and Validation MSEs, according to epochs, are shown in Fig. 8 and reveal the training processes of the networks. Because the overall training processes of the temperature and wind power data are similar, the training curves of the temperature data are only included in this paper. In general, the validation errors of DRNet-3 and fused DRNet-1 generally converge at between epoch 5 and 10. Therefore, the user-selectable early stopping strategy may be an option to expedite the training process when the validation error does not decrease for the predetermined epochs. However, this paper excludes the early stopping strategy to prevent early termination and to use the generally trained model.

C. Wind Power Forecasting Results

The wind power dataset is drawn from the hourly total wind output of ERCOT for 26311 hours from 2016 to 2018 [75]. The data displays the aggregated power output from all the wind generators in Texas. Total installed wind generation increased from 16246 MW to 22607 MW, and the maximum output of wind power was 19099 MW. The largest wind output percentage of the load was 54.6%, and the biggest percentage change of output was 280.6%. Among overall 26311 hours data, 76%, 16%, and 8% divisions are used for training, validation, and test set, respectively.

1) Residual Learnings: The first task of the case study is to examine WPF capability of the proposed residual learning. Each residual learning was applied to Bi-LSTM networks with the depth of 7 and 11, and all the activation functions are set as SeLU. Overall WPF results are shown in Fig. 9 and Table III. For 7-depth network, DRNet-4 outperforms the other methods. Among the conventional methods, the degradation for 11-depth

TABLE III
WPF PERFORMANCE OF PURE BI-LSTM NETWORK WITH VARIOUS RESIDUAL LEARNINGS

Depth	Residual Learning	Total Parameters	Average Time per Epoch (sec)	Training Error	Validation Error			Test Error		
				MSE ($\times 10^{-3}$)	MSE ($\times 10^{-3}$)	MAE ($\times 10^{-2}$)	MAPE (%)	MSE ($\times 10^{-3}$)	MAE ($\times 10^{-2}$)	MAPE (%)
7	MRN	54065	201.9	1.18117	1.24329	2.61980	9.11419	1.18727	2.52558	**10.26104**
	DenseNet	81969	195.7	1.24487	1.27612	2.66144	9.43301	1.17232	2.53073	10.79754
	DRNet-1	113553	199.4	1.21269	1.25607	2.63539	9.35526	**1.13836**	2.47652	10.47095
	DRNet-2	150481	202.1	1.20976	1.24331	2.62894	9.32037	1.16926	2.52468	10.53595
	DRNet-3	141361	202.9	1.19606	1.24451	2.62116	9.20104	1.14092	2.48192	10.31139
	DRNet-4	181361	206.2	**1.16272**	**1.24182**	**2.61316**	**9.09991**	1.14079	**2.46825**	10.30897
11	MRN	81265	331.6	1.31593	1.37610	2.78542	9.83265	1.36487	2.75777	11.53130
	DenseNet	238129	310.6	1.23790	1.26643	2.65208	9.55489	1.15059	2.49591	10.73446
	DRNet-1	187569	289.5	1.23054	1.27148	2.65067	9.33239	1.15231	2.50099	10.55659
	DRNet-2	305009	319.4	1.18092	1.24199	2.62390	9.21233	1.17948	2.51308	10.12464
	DRNet-3	308497	313.3	1.17695	**1.23464**	**2.60832**	**9.12849**	**1.11899**	**2.44881**	10.17813
	DRNet-4	400721	329.6	**1.16114**	1.24397	2.61758	9.14447	1.18986	2.51014	**10.01778**

TABLE IV
WPF PERFORMANCE OF LSTM AND BI-LSTM NETWORKS WITH DRNETS

Layers		Residual Learning	Total Parameters	Training Error	Validation Error			Test Error		
				MSE ($\times 10^{-3}$)	MSE ($\times 10^{-3}$)	MAE ($\times 10^{-2}$)	MAPE (%)	MSE ($\times 10^{-3}$)	MAE ($\times 10^{-2}$)	MAPE (%)
7	**7 LS**	DRNet-3	161841	**1.20351**	**1.25982**	**2.63709**	**9.10769**	**1.12553**	**2.45946**	**10.14412**
	7 Bi		141361	1.20711	1.26156	2.64669	9.34005	1.14719	2.49229	10.56358
	7 LS	DRNet-4	201841	1.19007	1.25889	2.64172	9.21327	1.16376	2.50653	10.41381
	5 LS, 2 Bi		191601	**1.18835**	**1.25452**	**2.63814**	9.30516	1.6654	2.50714	10.67459
	1 LS, 6 Bi		189553	1.19838	1.25934	2.64239	**9.19154**	**1.13319**	**2.46083**	**10.31093**
	7 Bi		181361	1.19228	1.25795	2.64284	9.48049	1.18828	2.53505	10.89894
11	11 LS	DRNet-3	337169	1.21182	1.28145	2.67182	9.32518	1.18351	2.52029	10.47664
	11 Bi		308497	**1.17695**	**1.23464**	**2.60832**	**9.12849**	**1.11899**	**2.44881**	**10.17813**
	11 LS	DRNet-4	429393	1.19753	1.26338	2.65131	9.35542	**1.18545**	2.53408	10.75092
	11 Bi		400721	**1.16114**	**1.24397**	**2.61758**	**9.14447**	1.15986	**2.51014**	**10.01708**

network does not occur in DenseNet only. The others performed well at 7-depth but a great increase of error can be observed at 11-depth network. On the other hand, DRNets showed superior forecasting accuracy to conventional methods. Though there is a slight increase in the error at 11-depth, DRNet-4 showed the lowest error at 7-depth. DRNet-2 and 3 not only showed good performance but also degradation did not occur. Especially, DRNets in 7-depth network showed higher accuracy than the 11-depth network with DenseNet, even with less number of parameters.

The better performance of DRNet-4 at 7-depth is owing to all the residual mappings in DRNet-4 containing spatiotemporal data, using both identity mapping and 1D CNN layer. However, overfitting occurred in 11-depth network with DRNet-4 as the number of parameters increases excessively. On the other hand, DRNet-3 has relatively high validation error in 7-depth, and the lowest error in 11-depth network. The difference between DRNet-3 and 4 is 1D CNN mapping of the initial input, which highly contributes to the increase of parameters. As shown in Table III and Fig. 9, the margin of parameter increase in DRNet-3 is much lower than one in DRNet-4, which leads to the prevention of overfitting or degradation. This can be identically stretched to DRNet-2 and 3 for deeper networks. In other words, the deeper the network is, the more 1D CNN mapping should be removed.

2) LSTM and Bi-LSTM Layers: The second task is to identify the impact of substituting Bi-LSTM for LSTM with DRNet-3 and 4. When the number of layers is 7, the network with 7 LSTM and one mixed with LSTM and Bi-LSTM showed the best performance each for DRNet-3 and DRNet-4, as shown in Table III. In case of 11 layers, the network with 11 Bi-LSTM surpassed the others. Fig. 10 shows the validation errors for various combinations of layers according to the network depth. 7-depth networks showed similar performance regardless of the type of the layers. However, the validation error of the 11-depth LSTM network increases with the increase of depth, while one of the pure Bi-LSTM network decreases. This is owing to the improvement in the parameter optimization of the backward layers in Bi-LSTM. As the network becomes deeper and more complex, Bi-LSTM layers can help to prevent overfitting and degradation, and to increase the accuracy of WPF.

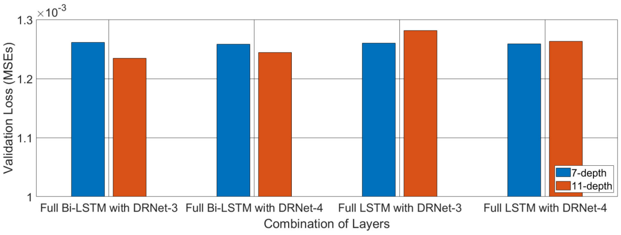


Fig. 10. Validation loss of various layer combinations with DRNets.

3) CNN Layer and Activation Functions: In order to verify the effectiveness of employing 1D CNN layer in the residual connection, DRNets and DRNets without 1D CNN layers are evaluated. In case of DRNets without 1D CNN layers, the residual connections remain with SeLU functions. According

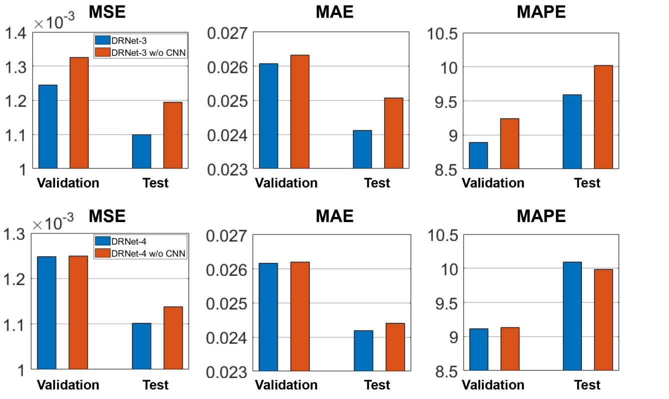


Fig. 11. Comparison of simulation results between DRNets and DRNets without 1D CNN.

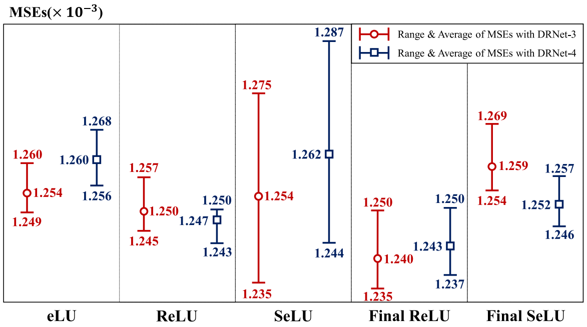


Fig. 12. Comparison of validation MSEs with DRNets for various activation functions.

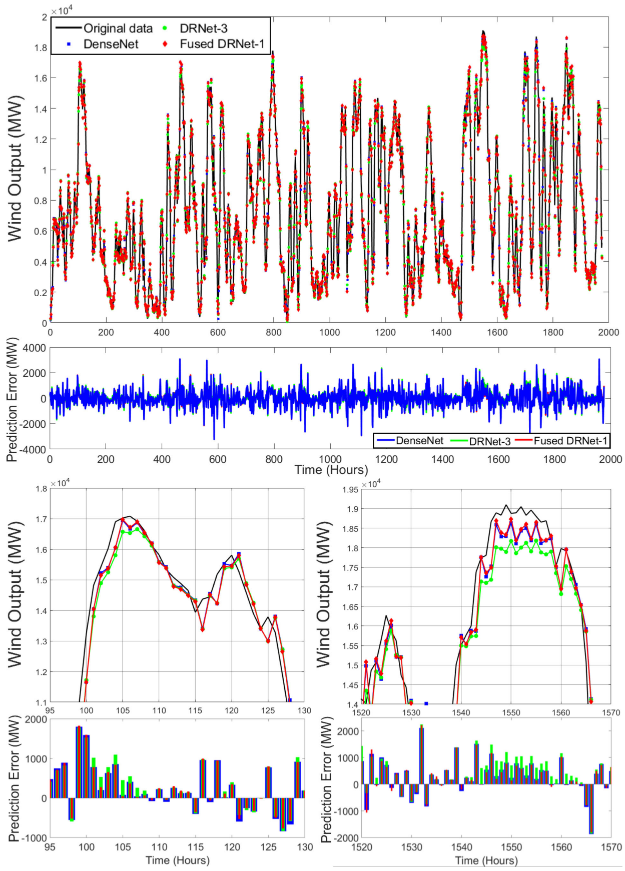


Fig. 13. WPF results and peak values forecasting of DenseNet, DRNet-3, and fused DRNet-1.

to the test result of Table III, 11-depth DRNet-3 and 7-depth DRNet-4 are used for comparison. The simulation results show that the DRNets with 1D CNN outperform DRNets without 1D CNN, as represented in Fig. 11. All the validation and test errors except the test MAPE of DRNet-4 are lower than those of DRNets without CNN. The next step is to determine the activation functions applied to 1D CNN and Dense layers. For the conventional activation functions, WPF results of the conventional functions, final ReLU, and final SeLU are compared. Final ReLU used SeLU for activating 1D CNN and ReLU for Dense layer and vice versa for final SeLU. Fig. 12 shows the comparison of validation MSEs according to activation functions. For conventional functions, ReLU has the lowest average error for both DRNet-3 and 4. However, SeLU recorded the lowest errors for specific experiments, though the deviation of the results was large. Final ReLU can nurture both strengths of ReLU and SeLU. The average MSEs of final ReLU were lower than those of ReLU. In addition, the best result showed better performance than one of SeLU. Overall deviations of final ReLU were larger than those of ReLU, but much lower than those of SeLU.

4) Fused Concept: WPF results of our best single model, i.e., 11-depth Bi-LSTM networks with DRNet-3 and final ReLU, are shown in Fig. 13. DRNet-3 showed better performance in forecasting low peak value than DenseNet, but has lower accuracy in forecasting high peak value. The fused concept can help to improve peak value forecasting. Therefore, the final model fused the short and long Bi-LSTM networks with DRNet-1, as the fused net highly increases the parameters. As shown in Fig. 13, fused DRNet-1 highly improves not only the high peak value forecasting but also the overall errors. The mean of the largest 10% errors for DRNet-3 was 275.2331 MW, which was higher than 247.0299 MW of DenseNet. Fused DRNet-1 improved these values to 199.4232 MW with almost the same standard deviation. Therefore, we can conclude that the Bi-LSTM forecasting model with final ReLU and fused DRNets enhances the overall WPF performance and peak value forecasting as it has strength in adjusting the data flow so that it can adequately optimize the parameters of DNN.

In order to verify the efficiency of our proposed models, Diebold-Mariano (DM) tests were conducted for ResNet, DenseNet, DRNet-3, and fused DRNet-1 [76], [77]. Results of DM tests based on the squared-error loss are shown in Table V . DM tests comparing ResNet to the other models show that the improvements made on DenseNet and DRNets are significant since the absolute values of DM are larger than 1.96, which is z score of 5% significance level in the normal distribution. Comparing DenseNet and DRNets, both DRNet-3 and Fused DRNet-1 have higher forecasting accuracy than DenseNet. The observed differences between DenseNet and Fused DRNet-1 are

TABLE V
RESULTS OF ABSOLUTE VALUE BASED DM TESTS

	ResNet & DenseNet	ResNet & DRNet-3	ResNet & Fused DRNet-1
DM	-2.5230	-4.3205	-3.0741
p-Value	0.0180	0.0002	0.0022
	DenseNet & DRNet-3	DenseNet & Fused DRNet-1	DRNet-3 & Fused DRNet-1
DM	-1.5051	-3.0460	-0.1058
p-Value	0.1310	0.0024	0.9124

TABLE VI
EVALUATION OF PROBABILISTIC FORECASTING METHODS

Forecasting Method	Proposed Method		SB Method	
	95%	99%	95%	99%
PICP (%)	92.51	96.81	85.16	88.08
PINAW (%)	12.86	16.92	7.69	10.12
CWC	57.5	67.5	1061.2	2389.3
Pinball Loss	10.8217		14.1298	
Average CRPS	0.0179		0.0201	

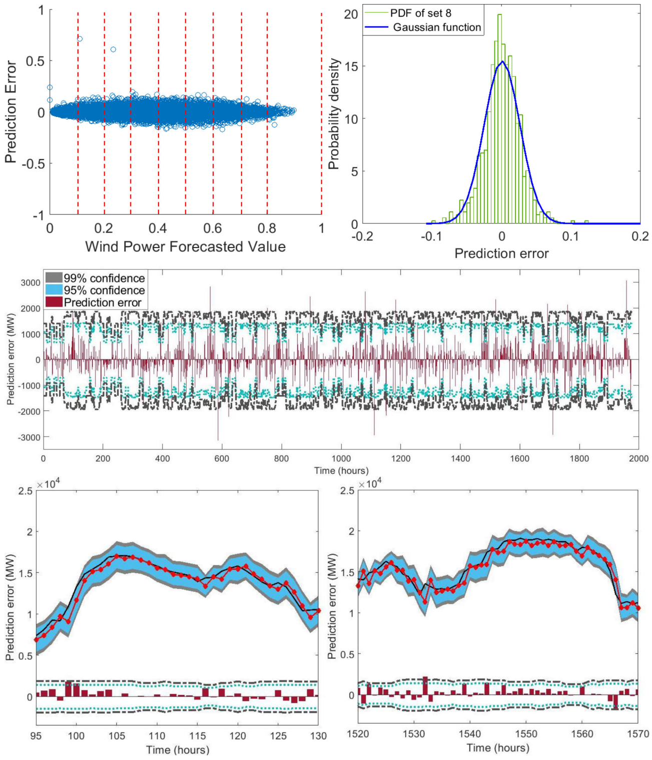


Fig. 14. Stochastic interval of WPF result with the fused model.

significant, with the absolute value of $\mathrm{DM} = 3.0460 > 1.96$. On the other hand, the absolute value of DM between DenseNet and DRNet-3 was 1.5051, which is less than 1.96. Therefore, the observed differences between DenseNet and DRNet-3 are not as significant as those between DenseNet and Fused DRNet-1, but still have meaningful value, because the value is higher than 1.282, which is z score of 10% level. DM test results on DRNet-3 and Fused DRNet-1 indicate that the forecasting accuracy of the two models can vary, according to the stochastic interference. However, it should be noticed that the test results represent the overall performance, not the peak value forecasting ability.

5) Stochastic Intervals: The prediction error of WPF with the fused model and the corresponding PIs with 95% and 99% confidence levels are represented in Fig. 14. Initially, the training prediction errors were divided into m sets in accordance with the forecasted values. The value of m is selected as 8, so that each set has more than 100 elements, i.e., $n(A_k) \geq 100$. Though independent random variables follow Gaussian distribution with $n(A_k) \geq 30$ by the central limit theorem, $n(A_k) \geq 100$ is chosen to get higher forecasting accuracy for more reliable power system operation [78]. Each set was transformed into a separate Gaussian distribution function, which is continuously updated with the test data. Note that the objective of the probabilistic forecasting in this paper is to get the range of the prediction error for test data. For given WPF value of test data from DNN, each data can be classified into A_k according to the forecasted value. PI for the test data is determined based on the mean and variance of the corresponding set, and the set is updated by merging the test data. For example, if the forecasted value of the first test data, $\hat{y}_1^{te}$, is included in A_j, PI about the forecasting error is determined as $[{L_1}^{\beta}(A_{j,1}), {U_1}^{\beta}(A_{j,1})]$, where $A_{j,1}$ is same with A_j. After the real value is revealed, the forecasting error, e_1^{te}, can be calculated and $A_{j,1}$ is updated to $A_{j,2}$, which contains $[\hat{y}_1^{te}, e_1^{te}]$ as a new element. Meanwhile, the other sets except for A_j remain the same as the previous.

In order to compare the performance of the proposed probabilistic forecasting to the standard bootstrap (SB) method, PI coverage probability (PICP), PI normalized averaged width (PINAW), and coverage width-based criterion (CWC) are adopted as performance indices [79]. Additionally, Pinball loss is calculated to evaluate the overall performance. Pinball loss can guarantee the probabilistic forecasting performance as a comprehensive index, which simultaneously evaluates the reliability, sharpness, and calibration. Physical meanings and mathematical equations of PICP, PINAW, and CWC can be found in [80], and Pinball loss used in this paper for certain β can be formulated as follows:

$$\text{Pinball Loss} = \sum_i \begin{cases} (\hat{W}_i^{te} - w_i^{te})\beta & \text{if } \hat{W}_i^{te} \geq w_i^{te}, \\ (w_i^{te} - \hat{W}_i^{te})(1-\beta) & \text{if } \hat{W}_i^{te} < w_i^{te}, \end{cases} \tag{19}$$

where estimated interval width, $\hat{W}_i^{te}$, and the difference between prediction error and historical prediction error mean, w_i^{te}, are defined as follows:

$$\hat{W}_i^{te} = z_{1-\beta/2}\sqrt{\sigma^2_{e_{B_n,i}}}, \tag{20}$$

$$w_i^{te} = |e_i - E[e_{B_{n,i}}]|. \tag{21}$$

Note that the Pinball loss provided in Table VI is the average of the Pinball values with $\beta = 0.01, 0.05, 0.1, 0.2, \ldots, 0.9$. Two hyperparameters in CWC are set to a value of 50 for η, and the confidence level, $1-\beta$, for μ. As shown in Table VI, the proposed probabilistic method shows higher PICP than the SB method for both 95% and 99% confidence level, which implies that the actual values lie in the proposed range with higher

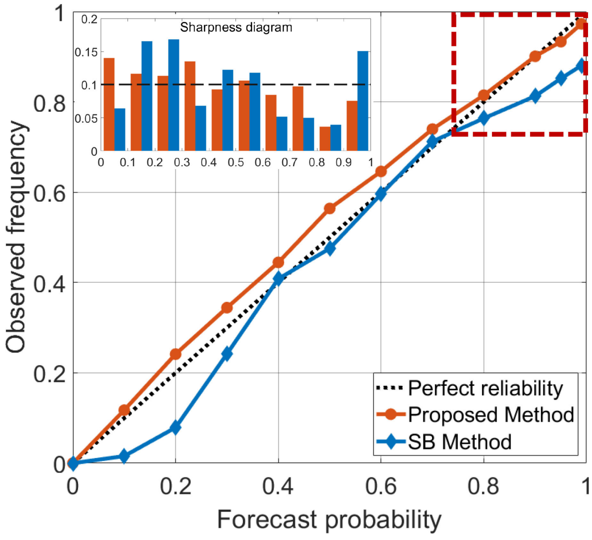


Fig. 15. Reliability diagram with the sharpness diagram of the probabilistic forecasting.

TABLE VII
WPF Performance Testing Using Various Forecasting Methods

Forecasting Method	Test Error		
	MSE ($\times10^{-3}$)	MAE ($\times10^{-2}$)	MAPE (%)
Naive Algorithm	20.313	11.598	34.797
ARIMA	9.4250	7.1411	29.270
Gaussian Process	1.8906	3.2035	11.797
SVM	1.5631	2.8301	11.730
3-Layer RNN	1.3211	2.5011	10.016
Fused DRNet-1	1.1251	2.4540	9.4041

possibility. Though PINAW of the proposed method is higher than one of the SB method, lower values of CWC and Pinball loss imply that the proposed method can derive more valid PI than the SB method.

The average continuous rank probability score (CRPS) in Table VI and the reliability diagram with sharpness in Fig. 15 affirm the high reliability of the proposed method. The reliability diagram of the proposed method is more adjacent to the perfect reliability curve than one of the SB methods, especially for the forecast probability larger than 0.8. As remarked in [81], [82], small deviations from the diagonal with small bias in the sharpness verify the reliability of the proposed method. In addition, smaller CRPS of the proposed method than the SB method indicates the adequacy of the Gaussian distributions postulated from the proposed method: the CRPS calculation for the Gaussian distribution is well documented in [83].

6) Forecasting Performance Comparison: The WPF performance of the proposed method is compared to other methods, including the naive algorithm, ARIMA, Gaussian process (GP), Support Vector Machine (SVM), and 3-layer RNN model and is summarized in Table VII. The naive algorithm is one of the simple physical forecasting methods, and the combination of a seasonal and hourly naive algorithms is used for comparison [84]. As one of the conventional statistical methods, ARIMA is used, where the structure is determined to be ARIMA(2,0,1) based on the autocorrelation and partial autocorrelation plots [85]. The GP and SVM are representative examples of shallow machine learning methods. For the GP, Matern and White kernel are combined within a Bayesian framework, and support vector regression model is used for the SVM [86], [87]. The 3-layer RNN represents a relatively small size of ANN.

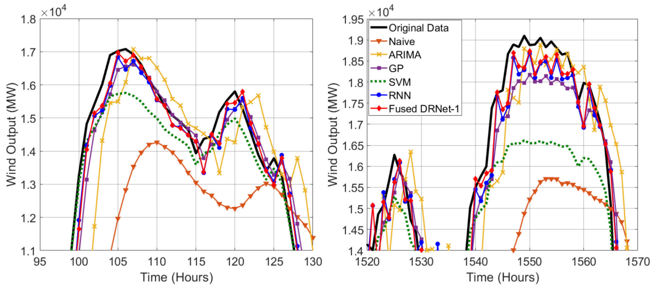
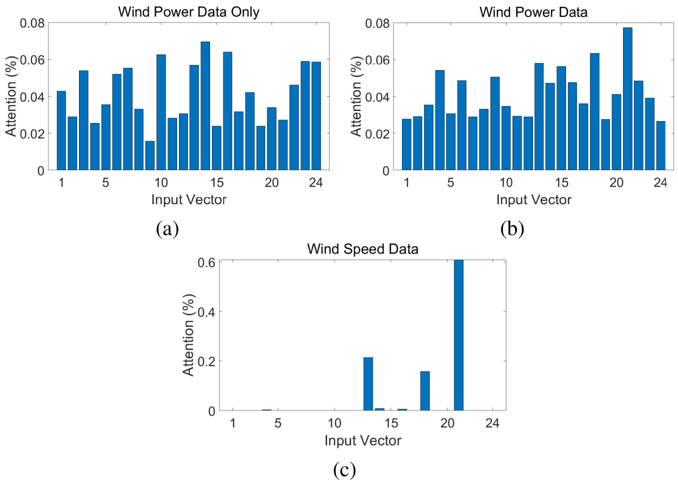


Fig. 16. WPF profiles using various forecasting methods.

As shown in Table VII, the machine learning methods excel the physical and conventional statistical methods for all three metrics. The DM values of GP with the naive algorithm and ARIMA are -12.9265 and -11.7738 each. Thus, the GP shows even higher performance than the naive algorithm and ARIMA. At the same time, SVM and 3-layer RNN show lower test errors than GP. DM values of SVM and RNN with GP are -6.0173 and -8.9571, which denotes the meaningful observed differences. Though RNN performs slightly better than SVM, there are no significant differences between the two models because DM value between SVM and RNN is 0.3362. It is remarkable that the proposed model has improved the forecasting accuracy of the SVM and 3-Layer RNN significantly. DM values of fused DRNet-1 with SVM and RNN are -2.5514 and -3.0518. Forecasted wind power profiles using all methods are shown in Fig. 16. It is noteworthy that the profiles with lower accuracy show longer time delay or more massive peak value error than those with higher performance.

D. Additional Use of Wind Speed Data

In order to investigate the impact of incorporating weather data explicitly, for example, wind speed data as an input, additional experiments on the wind speed and the ERCOT wind power output data from 2016 to 2017 are conducted. Wind power capacities by site in ERCOT are obtained from [88] and wind speed data is drawn from [89]. The actual wind speed input set for the aggregated WPF is obtained by calculating a weighted arithmetic mean of hourly wind speeds by regional wind power capacities with reference to the total capacity. For the impact analysis, the global attention mechanism is adopted and the location-based attention scores are compared as detailed in [90]. The WPF test results based on attention mechanism and wind speed data are shown in Table VII. The attention scores are represented in Fig. 17 where the x-axis indicates the input vector with 24 hours of length, and the attention scores on y-axis mean the average of intensified weights for the vector at a specific hour. When the attention mechanism is applied to the network only



(a)

(b)

(c)

Fig. 17. Attention scores of (a) the network only with wind power data, and the scores of (b) the wind power data, and (c) wind speed data of the network with both data.

TABLE VIII
IMPACT OF WIND SPEED DATA AND ATTENTION MECHANISM ON WPF TEST RESULTS

Wind Speed Data	Attention Mechanism	Test Error		
		MSE ($\times10^{-3}$)	MAE ($\times10^{-2}$)	MAPE (%)
Without Data	Without Mechanism	1.6728	2.9634	9.3701
	With Mechanism	1.6451	2.9391	9.2909
With Data	Without Mechanism	2.7863	3.9495	17.048
	With Mechanism	8.6707	7.2192	20.398

with wind power data, the accuracy of the network increases, as indicated in Table VIII. This is owing to the attention weights, which increase the impact of the more relevant inputs.

On the other hand, the accuracy of the network using both wind power and wind speed data decreases with the attention. Though the scores of the wind power are similar to those only using wind power data, the scores of 3 hours, 6 hours, and 11 hours ahead wind speed data have turned out to be much higher than the others. Therefore, the performance of the network using both data is dominated by wind speed data at specific points, rather than the sequence of wind power or wind speed. The following observations are then drawn from the analysis:

1) Though the raw wind speed data is obtained from a reliable source and has been preprocessed to be an adequate pair with the wind power data, the wind speed was not measured at the same sites for the aggregate wind power forecasting. There are also unknown (or unexplainable) dynamics on top of the underlying physics between the wind speed and output power, which indeed requires an even higher dimensional model [91]–[93].
2) Incorporating the wind speed data thus increases the uncertainty and may degrade the overall performance. Focusing on the wind power sequence with attention mechanism draws higher performance for the 1-h ahead WPF as the wind power data assumes all of the dynamics the proposed model tries to identify.
3) A high attention score confirms the significant relationship between wind power and speed. Instead of being used as a direct input to the forecasting model, the wind speed data may be used as an auxiliary signal or indicator to improve the WPF performance.

V. Conclusion

This paper proposed the deep learning model for 1-h ahead WPF, where the basic layer is composed of Bi-LSTM layer. Bi-LSTM network has been widely investigated as a powerful forecasting model as it can improve performance by eliminating propagated errors, especially when the number of parameters increases by residual learnings. The increasing depth of DNN, however, makes the LSTM prone to overfitting, which degrades the performance of the deep learning model. The proposed DRNets effectively resolve this technical challenge by concatenating the original input, shortcuts of the residuals, and the original activated input. The proposed activation functions using SeLU for 1D CNN and ReLU for Dense layer can further improve the overall accuracy. Significant improvements in the peak value forecasting have been observed in the case study by using the fused network of short and long Bi-LSTM networks with DRNets. Consistent superior and reliable performance of the proposed model for various datasets demonstrates that the proposed method provides a general framework for time-series forecasting applications, especially in grid power operations.

References

[1] E. Du *et al.*, “Operation of a high renewable penetrated power system with CSP plants: A. look-ahead stochastic unit commitment model,” *IEEE Trans. Power Syst.*, vol. 34, no. 1, pp. 140–151, Jan. 2019.
[2] B. Kroposki *et al.*, “Achieving a 100% renewable grid: Operating electric power systems with extremely high levels of variable renewable energy,” *IEEE Power Energy Mag.*, vol. 15, no. 2, pp. 61–73, Mar./Apr. 2017.
[3] M. Khodayar, S. Mohammadi, M. E. Khodayar, J. Wang, and G. Liu, “Convolutional graph autoencoder: A generative deep neural network for probabilistic spatio-temporal solar irradiance forecasting,” *IEEE Trans. Sustain. Energy*, vol. 11, no. 2, pp. 571–583, Apr. 2020.
[4] M. Marinelli, P. Maule, A. N. Hahmann, O. Gehrke, P. B. Norgrd, and N. A. Cutululis, “Wind and photovoltaic large-scale regional models for hourly production evaluation,” *IEEE Trans. Sustain. Energy*, vol. 6, no. 3, pp. 916–923, Jul. 2015.
[5] M. F. Tahir, C. Haoyong, A. Khan, M. S. Javed, N. A. Laraik, and K. Mehmood, “Optimizing size of variable renewable energy sources by incorporating energy storage and demand response,” *IEEE Access*, vol. 7, pp. 103 115–103 126, 2019.
[6] A. Dhabi, IRENA, “Renewable energy statistics 2020,” The internation renewable energy agency, Abu Dhabi, UAE, Jul. 2020. [Online]. Available: https://www.irena.org/publications/2020/Jul/Renewable-energy-statistics-2020
[7] L. Bird, M. Milligan, and D. Lew, “Integrating variable renewable energy: Challenges and solutions,” Nat. Renewable Energy Lab. (NREL), Golden, CO, USA, Tech. Rep., 2013.
[8] Z. Zhuo *et al.*, “Transmission expansion planning test system for AC/DC hybrid grid with high variable renewable energy penetration,” *IEEE Trans. Power Syst.*, vol. 35, no. 4, pp. 2597–2608, Jul. 2020.
[9] M. Nehrir *et al.*, “A review of hybrid renewable/alternative energy systems for electric power generation: Configurations, control, and applications,” *IEEE Trans. Sustain. Energy*, vol. 2, no. 4, pp. 392–403, Oct. 2011.
[10] L. Yang, M. He, J. Zhang, and V. Vittal, “Support-vector-machine-enhanced Markov model for short-term wind power forecast,” *IEEE Trans. Sustain. Energy*, vol. 6, no. 3, pp. 791–799, Jul. 2015.

[11] M. H. Rehmani, M. Reisslein, A. Rachedi, M. Erol-Kantarci, and M. Radenkovic, “Integrating renewable energy resources into the smart grid: Recent developments in information and communication technologies,” *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 2814–2825, Jul. 2018.

[12] Iea, Distributed solar PV, 2019. [Online]. Available: https://www.iea.org/reports/renewables-2019/distributed-solar-pv#abstract

[13] A. Orrel, D. Preziuso, N. Foster, S. Morris, and J. Homer, 2018 Distributed Wind Market Report, 2018. [Online]. Available: https://www.energy.gov/eere/wind/downloads/2018-distributed-wind-market-report

[14] L. Exizidis, J. Kazempour, P. Pinson, Z. De Grève, and F. Vallée, “Impact of public aggregate wind forecasts on electricity market outcomes,” *IEEE Trans. Sustain. Energy*, vol. 8, no. 4, pp. 1394–1405, Oct. 2017.

[15] M. Jia, C. Shen, and Z. Wang, “A distributed probabilistic modeling algorithm for the aggregated power forecast error of multiple newly built wind farms,” *IEEE Trans. Sustain. Energy*, vol. 10, no. 4, pp. 1857–1866, Oct. 2019.

[16] P. Mandal, H. Zareipour, and W. D. Rosehart, “Forecasting aggregated wind power production of multiple wind farms using hybrid wavelet-psonns,” *Int. J. Energy Res.*, vol. 38, no. 13, pp. 1654–1666, 2014.

[17] Y.-L. Hu and L. Chen, “A nonlinear hybrid wind speed forecasting model using LSTM network, hysteretic ELM and differential evolution algorithm,” *Energy Convers. Manage.*, vol. 173, pp. 123–142, 2018.

[18] H. Liu, X.-W. Mi, and Y.-F. Li, “Wind speed forecasting method based on deep learning strategy using empirical wavelet transform, long short term memory neural network and elman neural network,” *Energy Convers. Manage.*, vol. 156, pp. 498–514, 2018.

[19] U. K. Das *et al.*, “Forecasting of photovoltaic power generation and model optimization: A review,” *Renewable Sustain. Energy Rev.*, vol. 81, pp. 912–928, 2018.

[20] H. A. Nielsen, T. S. Nielsen, H. Madsen, M. J. S. I. Pindado, and I. Marti, “Optimal combination of wind power forecasts,” *Wind Energy: Int. J. Prog. Appl. Wind Power Convers. Technol.*, vol. 10, no. 5, pp. 471–482, 2007.

[21] M. Lange and U. Focken, “Physical approach to short-term wind power prediction,” Berlin, Germany: Springer, 2006, vol. 208.

[22] G. Giebel, R. Brownsword, G. Kariniotakis, M. Denhard, and C. Draxl, “State-of-the-art in short-term prediction of wind power: A literature overview,” *ANEMOS. plus*, 2011. [Online]. Available: https://www.osti.gov/etdeweb/servlets/purl/1011554

[23] Y. Ren, P. Suganthan, and N. Srikanth, “Ensemble methods for wind and solar power forecasting-a state-of-the-art review,” *Renewable Sustain. Energy Rev.*, vol. 50, pp. 82–91, 2015.

[24] C. Zhang, J. Zhou, C. Li, W. Fu, and T. Peng, “A compound structure of ELM based on feature selection and parameter optimization using hybrid backtracking search algorithm for wind speed forecasting,” *Energy Convers. Manage.*, vol. 143, pp. 360–376, 2017.

[25] O. A. Maatallah, A. Achuthan, K. Janoyan, and P. Marzocca, “Recursive wind speed forecasting based on hammerstein auto-regressive model,” *Appl. Energy*, vol. 145, pp. 191–197, 2015.

[26] A. A. Ezzat, “Turbine-specific short-term wind speed forecasting considering within-farm wind field dependencies and fluctuations,” *Appl. Energy*, vol. 269, 2020, Art. no. 115034.

[27] P. Pinson, L. Christensen, H. Madsen, P. E. Sorensen, M. H. Donovan, and L. E. Jensen, “Regime-switching modelling of the fluctuations of offshore wind generation,” *J. Wind Eng. Ind. Aerodyn.*, vol. 96, no. 12, pp. 2327–2347, 2008.

[28] E. Cadenas and W. Rivera, “Short term wind speed forecasting in la venta, Oaxaca, México, using artificial neural networks,” *Renewable Energy*, vol. 34, no. 1, pp. 274–278, 2009.

[29] J. P. d. S. Catalão, H. M. I. Pousinho, and V. M. F. Mendes, “Short-term wind power forecasting in portugal by neural networks and wavelet transform,” *Renewable Energy*, vol. 36, no. 4, pp. 1245–1251, 2011.

[30] H. Liu, H.-Q. Tian, and Y.-F. Li, “Comparison of two new arima-ANN and arima-Kalman hybrid methods for wind speed prediction,” *Appl. Energy*, vol. 98, pp. 415–424, 2012.

[31] C. Feng, M. Cui, B.-M. Hodge, and J. Zhang, “A data-driven multi-model methodology with deep feature selection for short-term wind forecasting,” *Appl. Energy*, vol. 190, pp. 1245–1257, 2017.

[32] Q. Cao, B. T. Ewing, and M. A. Thompson, “Forecasting wind speed with recurrent neural networks,” *Eur. J. Oper. Res.*, vol. 221, no. 1, pp. 148–154, 2012.

[33] T. Senjyu, A. Yona, N. Urasaki, and T. Funabashi, “Application of recurrent neural network to long-term-ahead generating power forecasting for wind power generator,” in *Proc. IEEE PES Power Syst. Conf. Expo.*, 2006, pp. 1260–1265.

[34] Q. Xiaoyun, K. Xiaoning, Z. Chao, J. Shuai, and M. Xiuda, “Short-term prediction of wind power based on deep long short-term memory,” in *Proc. IEEE PES Asia-Pacific Power Energy Eng. Conf.*, 2016, pp. 1148–1152.

[35] M. Khodayar and J. Wang, “Spatio-temporal graph deep neural network for short-term wind speed forecasting,” *IEEE Trans. Sustain. Energy*, vol. 10, no. 2, pp. 670–681, Apr. 2019.

[36] H. Liu, X. Mi, and Y. Li, “Smart multi-step deep learning model for wind speed forecasting based on variational mode decomposition, singular spectrum analysis, LSTM network and ELM,” *Energy Convers. Manage.*, vol. 159, pp. 54–64, 2018.

[37] S. Li, P. Wang, and L. Goel, “Wind power forecasting using neural network ensembles with feature selection,” *IEEE Trans. Sustain. Energy*, vol. 6, no. 4, pp. 1447–1456, Oct. 2015.

[38] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[39] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[40] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional LSTM networks,” in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2005, pp. 2047–2052.

[41] C. Szegedy *et al.*, “Going deeper with convolutions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[42] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.

[43] H. Jahangir, H. Tayarani, S. S. Gougheri, M. A. Golkar, A. Ahmadian, and A. Elkamel, “Deep learning-based forecasting approach in smart grids with micro-clustering and bi-directional lstm network,” *IEEE Trans. Ind. Electron.*, to be published, doi: 10.1109/TIE.2020.3009604.

[44] J.-F. Toubeau, J. Bottieau, F. Vallée, and Z. De Grève, “Deep learning-based multivariate probabilistic forecasting for short-term scheduling in power markets,” *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 1203–1215, Mar. 2019.

[45] ERCOT “High level overview of ERCOT wind power forecasting process,” ERCOT, Texas, USA, 2020. [Online]. Available: https://mis.ercot.com/misapp/GetReports.do?reportTypeID=19375&reportTitl%e=IRR%20Forecasting%20Process&showHTMLView=&mimicKey

[46] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[47] H. Shi, M. Xu, and R. Li, “Deep learning for household load forecasting-a novel pooling deep RNN,” *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5271–5280, Sep. 2018.

[48] A. Graves, A.-R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2013, pp. 6645–6649.

[49] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, “Short-term residential load forecasting based on LSTM recurrent neural network,” *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan. 2019.

[50] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[51] I. Goodfellow, Y. Bengio, and A. Courville, “Deep learning,” Cambridge, MA, USA: MIT press, 2016.

[52] T. Fukada, M. Schuster, and Y. Sagisaka, “Phoneme boundary estimation using bidirectional recurrent neural networks and its applications,” *Syst. Comput. Jpn.*, vol. 30, no. 4, pp. 20–30, 1999.

[53] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.

[54] N.Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[55] Y.-L. Boureau, J. Ponce, and Y. LeCun, “A theoretical analysis of feature pooling in visual recognition,” in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 111–118.

[56] Y. Gal and Z. Ghahramani, “A theoretically grounded application of dropout in recurrent neural networks,” in *Adv. Neural Inf. Process. Syst.*, 2016, pp. 1019–1027.

[57] W. Zaremba, I. Sutskever, and O. Vinyals, “Recurrent neural network regularization,” 2015, *arXiv:1409.2329*.

[58] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, “A fully trainable network with RNN-based pooling,” *Neurocomputing*, vol. 338, pp. 72–82, 2019.

[59] H. T. Siegelmann and E. D. Sontag, “On the computational power of neural nets,” in *Proc. 5th Annu. Workshop Comput. Learn. Theory*, 1992, pp. 440–449.

[60] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-ResNet and the impact of residual connections on learning,” in *Proc. 31st AAAI Conf. Artif. Intell.*, 2016, *arXiv:1602.07261*.

[61] M. Abdi and S. Nahavandi, “Multi-residual networks: Improving the speed and accuracy of residual networks,” 2017, *arXiv:1609.05672*.

[62] L. Zhao, J. Wang, X. Li, Z. Tu, and W. Zeng, “On the connection of deep fusion to ensembling,” 2016, *arXiv:1611.07718*.

[63] K. Zhang, M. Sun, T. X. Han, X. Yuan, L. Guo, and T. Liu, “Residual networks of residual networks: Multilevel residual networks,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 6, pp. 1303–1314, Jan. 2017.

[64] Z. Yu, Z. Niu, W. Tang, and Q. Wu, “Deep learning for daily peak load forecasting-A novel gated recurrent neural network combining dynamic time warping,” *IEEE Access*, vol. 7, pp. 17 184–17 194, 2019.

[65] L. M. Saini and M. K. Soni, “Artificial neural network-based peak load forecasting using conjugate gradient methods,” *IEEE Trans. Power Syst.*, vol. 17, no. 3, pp. 907–912, Aug. 2002.

[66] N. Amjady, “Short-term hourly load forecasting using time-series modeling with peak load estimation capability,” *IEEE Trans. Power Syst.*, vol. 16, no. 3, pp. 498–505, Aug. 2001.

[67] G. E. Dahl, T. N. Sainath, and G. E. Hinton, “Improving deep neural networks for lvcsr using rectified linear units and dropout,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2013, pp. 8609–8613.

[68] X. Zhang, Y. Zou, and W. Shi, “Dilated convolution neural network with leakyrelu for environmental sound classification,” in *Proc. 22nd Int. Conf. Digit. Signal Process.*, Aug. 2017, pp. 1–5.

[69] D. Pedamonti, “Comparison of non-linear activation functions for deep neural networks on MNIST classification task,” 2018, *arXiv:1804.02763*.

[70] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation functions: Comparison of trends in practice and research for deep learning,” CoRR, vol. abs/1811.03378, 2018. [Online]. Available: http://arxiv.org/abs/1811.03378

[71] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong, “Probabilistic forecasting of wind power generation using extreme learning machine,” *IEEE Trans. Power Syst.*, vol. 29, no. 3, pp. 1033–1044, May 2014.

[72] K. Bruninx and E. Delarue, “A statistical description of the error on wind power forecasts for probabilistic reserve sizing,” *IEEE Trans. Sustain. Energy*, vol. 5, no. 3, pp. 995–1002, Jul. 2014.

[73] E. Bisong, “Introduction to scikit-learn,” in *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Berlin, Germany: Springer, 2019, pp. 215–229.

[74] Stytch, Jena’s Climate Change, 2018. [Online]. Available: https://www.kaggle.com/stytch16/jena-climate-2009-2016

[75] ERCOT, “Intermittent renewable resources,” 2019. [Online]. Available: http://www.ercot.com/gridinfo/generation

[76] F. X. Diebold and R. S. Mariano, “Comparing predictive accuracy,” *J. Bus. Econ. Statist.*, vol. 20, no. 1, pp. 134–144, 2002.

[77] H. Chen, Q. Wan, and Y. Wang, “Refined diebold-mariano test methods for the evaluation of wind power forecasting models,” *Energies*, vol. 7, no. 7, pp. 4185–4198, 2014.

[78] R. Durrett, “Probability: Theory and examples,” Cambridge, U.K.: Cambridge Univ. Press, 2019.

[79] R. W. Johnson, “An introduction to the bootstrap,” *Teach. Statist.*, vol. 23, no. 2, pp. 49–54, 2001.

[80] 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.

[81] J. Bröcker and L. A. Smith, “Increasing the reliability of reliability diagrams,” *Weather Forecasting*, vol. 22, no. 3, pp. 651–661, 2007.

[82] R. Perez, “Wind field and solar radiation characterization and forecasting: A numerical approach for complex terrain,” Berlin, Germany: Springer, 2018.

[83] T. Gneiting and A. E. Raftery, “Strictly proper scoring rules, prediction, and estimation,” *J. Amer. Stat. Assoc.*, vol. 102, no. 477, pp. 359–378, 2007.

[84] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, Melbourne, Australia: OTexts, 2018.

[85] F. A. Eldali, T. M. Hansen, S. Suryanarayanan, and E. K. Chong, “Employing arima models to improve wind power forecasts: A. case study in ercot,” in *Proc. IEEE North Amer. Power Symp.*, 2016, pp. 1–6.

[86] V. Borovitskiy, A. Terenin, P. Mostowsky, and M. P. Deisenroth, “Matern Gaussian processes on riemannian manifolds,” 2020, *arXiv:2006.10160*.

[87] Y. Chen *et al.*, “Short-term electrical load forecasting using the support vector regression (svr) model to calculate the demand response baseline for office buildings,” *Appl. Energy*, vol. 195, pp. 659–670, 2017.

[88] ERCOT, ERCOT Wind Patterns for Existing Sites, 2018. [Online]. Available: http://www.ercot.com/gridinfo/resource

[89] NREL, NREL National Solar Radiation Database (NSRDB), Dec. 2019. [Online]. Available: https://sam.nrel.gov/weather-data

[90] M.-T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” 2015, *arXiv:1508.04025*.

[91] Y. Wang, Q. Hu, and S. Pei, “Wind power curve modeling with asymmetric error distribution,” *IEEE Trans. Sustain. Energy*, vol. 11, no. 3, pp. 1199–1209, Jul. 2020.

[92] J. Yan, H. Zhang, Y. Liu, S. Han, and L. Li, “Uncertainty estimation for wind energy conversion by probabilistic wind turbine power curve modelling,” *Appl. Energy*, vol. 239, pp. 1356–1370, 2019.

[93] Y. Zhao, L. Ye, W. Wang, H. Sun, Y. Ju, and Y. Tang, “Data-driven correction approach to refine power curve of wind farm under wind curtailment,” *IEEE Trans. Sustain. Energy*, vol. 9, no. 1, pp. 95–105, Jan. 2018.

Min-Seung Ko (Student Member, IEEE) received the B.S. degree in electrical engineering in 2018 from Yonsei University, Seoul, South Korea, where he has been working toward the M.S./Ph.D. degrees since September 2018. His research interests include integration of variable renewable energy sources, power system operation and voltage control, and applications of deep learning and optimization in power systems.

Kwangsuk Lee (Member, IEEE) received the B.S. and M.S. degrees and completed the Doctoral course in electrical engineering from Yonsei University, Seoul, South Korea. From 2006 to 2010, he was a Researcher with Yonsei University Automation Research Center. Since 2010, he has been with SK Telecom, Seoul, Korea, as a Data Analyst. His research interests include anomaly detection, fault diagnosis, estimation of remaining useful life, pruning, quantization, model compression, model optimization, machine learning, and deep learning.

Jae-Kyeong Kim (Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from Yonsei University, Seoul, South Korea, in 2012 and 2018, respectively. In 2018, he was a Postdoctoral Researcher with Yonsei University. Since December 2018, he has been with Korea Electrotechnology Research Institute (KERI), Uiwang, Korea. His research interests include power system modeling, voltage stability, power system dynamics, impacts of parametric uncertainty, and applications of deep learning and optimization.

Chang Woo Hong (Graduate Student Member, IEEE) received the B.S. degree in foreign language from the Republic of Korea Naval Academy, Jinhae, South Korea, in 2006, and the M.S. degrees in mechanical engineering from Yonsei University, Seoul, South Korea, in 2016. He is currently working toward the Ph.D. degree with the Department of Electrical and Electronic Engineering, Yonsei University, Seoul, South Korea. From 2006, he was with ROK Navy, specializing in engineering and logistics. His current research interests include an electric ship, dc grids and load forecasting, and PHM (Prognostics and Health Management) of the military application.

Zhao Yang Dong (Fellow, IEEE) is a Professor of Energy Systems with the University of New South Wales (UNSW) Australia. He is also the Director of UNSW Digital Grid Futures Institute and ARC Research Hub for Integrated Energy Storage Solutions. His research interest includes power system planning and stability, smart grid/micro-grid, load modeling, electricity market, and smart city planning. He was the Ausgrid Chair and Director of Ausgrid Centre for Intelligent Electricity Networks to provide R&D support for the Smart Grid, Smart City national demonstration project. He has been a Editor for several IEEE TRANSACTIONS and IET journals. He is a Web of Science highly cited Researcher since 2019.

Kyeon Hur (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Yonsei University, Seoul, South Korea, in 1996 and 1998, respectively, and the Ph.D. degree in electrical and computer engineering from The University of Texas at Austin in 2007.

He was an R&D engineer with Samsung Electronics, Suwon, South Korea, between 1998 and 2003, where he designed control algorithms and power-electronic circuits for AC drives. His industrial experience includes the Electric Reliability Council of Texas (ERCOT), Taylor, TX, USA as a Grid Operations Engineer between 2007 and 2008. He was also with the Electric Power Research Institute (EPRI), Palo Alto, CA, USA and conducted and managed research projects in Grid Operations and Planning from 2008 to 2010. He has rejoined Yonsei University since 2010 and leads a smart-grid research group. His current research interests include FACTS/HVDC, PMU-based analysis and control, integration of variable generation and controllable load, and load modeling.