Seasonal Self-evolving Neural Networks Based Short-term Wind Farm Generation Forecast

Yunchuan Liu*, Amir Ghasemkhani†, Lei Yang*, Jun Zhao‡, Junshan Zhang§, and Vijay Vittal§
* Department of Computer Science and Engineering, University of Nevada, Reno, NV, 89509
† School of Computer Science and Engineering, California State University San Bernardino, San Bernardino, CA, 92407
‡ School of Computer Science and Engineering, Nanyang Technological University, Singapore, 639798
§School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, 85287

***Abstract*—This paper studies short-term wind farm generation forecast. From the real wind generation data, we observe that wind farm generation exhibits the non-stationarity and the seasonality and that the dynamics of non-ramp, ramp-up, and ramp-down events are different across different classes of wind turbines. To deal with such heterogeneous dynamics of wind farm generation, we propose seasonal self-evolving neural networks based short-term wind farm generation forecast. The proposed approach first classifies the historical data into ramp-up and ramp-down datasets and non-ramp datasets for different seasons, and then trains different neural networks for each dataset to capture different wind farm power dynamics. To account for the non-stationarity as well as reduce the burden of hyperparameter tuning, we leverage NeuroEvolution of Augmenting Topologies (NEAT) to train neural networks, which evolves the neural networks using a genetic algorithm to find the best weighting parameters and network topology. Based on the proposed seasonal self-evolving neural networks, we develop algorithms for both point forecasts and distributional forecasts. Experimental results, using the real wind generation data, demonstrate the significantly improved accuracy of the proposed forecast approach, compared with other forecast approaches.**

***Index Terms*—Short-term wind power forecast, Self-evolving neural networks, Point forecast, Distributional forecast.**

I. Introduction

Imputable to the diminishing traditional energy and environmental contamination, much effort has been invested to integrate renewable energy sources, such as wind and solar. Indeed, wind energy constitutes a significant portion of this renewable integration [1]. Due to the high variability and difficult-to-control dynamics of wind energy, high penetration of wind generation brings imperative operational challenges, especially during wind power ramps. This requires accurate forecasts of future wind generation. In this paper, we aim to develop accurate forecast approaches for wind farm generation, eminently for wind power ramps.

There have been many studies on wind generation forecast based on physical models (e.g., numerical weather prediction model [2]) and statistical models (e.g., autoregressive model [3], Markov chains [4]). Recently, advanced artificial intelligence (AI) techniques have been successfully applied to many applications. There have been some studies applying AI techniques for wind generation forecast (e.g., support vector machine (SVM) [5], Artificial Neural Networks (ANN) [6], Wavelet Neural Network (WNN) [7], Adaptive Neuro-Fuzzy Neural Network (ANFIS) [8]). In [9], a long short-term memory (LSTM) prediction model is proposed for short-term wind power forecast. In [10], a two-stage forecasting model based on the error factor and the ensemble method is proposed for multi-step wind power forecast. To deal with the non-stationarity of wind generation, wavelet components decomposition [5], empirical mode decomposition (EMD) based model [11], and variational mode decomposition (VMD) based model [12] are introduced; however, for these works, it is challenging to find appropriate numbers of components or modes.

Although neural network based methods may improve the prediction performance to some extent, it has been shown that these methods may have poor performance if extreme (or ramp) events are overlooked [13]. Our previous works [14], [15] have shown 1) the non-stationarity and the seasonality of wind farm generation and 2) different dynamics of non-ramp, ramp-up, and ramp-down events of wind farm generation. Therefore, simply applying neural networks without considering ramp events may not obtain the best prediction performance (see the experimental results in Sec. IV), not to mention the tuning of network topology and hyperparameters, which is often a demanding task. Moreover, as wind farms often consist of different classes of wind turbines, the dynamics of wind generation of different classes of wind turbines can be different, which is observed in our dataset (see Sec. II). Motivated by these observations, we aim to develop neural networks based wind generation forecast approaches that consider these unique features of wind farm generation.

In this paper, we propose seasonal self-evolving neural networks based short-term wind farm generation forecast that accounts for different power output dynamics of different wind turbines under non-ramp, ramp-up, and ramp-down events, in order to achieve a more accurate wind farm generation forecast. The basic idea is as follows: 1) first classify the historical data into non-ramp, ramp-up, and ramp-down datasets, where the non-ramp dataset is further split into 4 datasets for 4 seasons to account for the seasonality; and 2) then develop

This work was supported in part by the U.S. National Science Foundation under Grants EEC-1801727, IIS-1838024, and CNS-1950485

978-1-7281-6127-3/20/$31.00

different neural networks for different power output dynamics of different turbines under non-ramp, ramp-up, and ramp-down events, with the intention of accounting for the unique features of wind farm generation. To account for the non-stationarity as well as reduce the burden of hyperparameter tuning, we leverage NeuroEvolution of Augmenting Topologies (NEAT) [16] to train neural networks, which evolves the neural networks using a genetic algorithm to find the best weighting parameters and network topology. Based on the proposed seasonal self-evolving neural networks, we study both point forecasts and distributional forecasts. Based on wind measurement data from a real-world wind farm, the proposed forecast approach demonstrates significantly improved forecast accuracy, compared with other approaches. Compared to the existing works, this work provides a unified framework that considers 1) the non-stationarity and the seasonality of wind farm power outputs and 2) different dynamics of wind non-ramp, ramp-up, and ramp-down events across different classes of turbines. By using self-evolving neural networks, the proposed approach does not require AI experts to tune the topology of neural networks and the hyperparameters. Thus, the proposed approach can be easily implemented in practice.

The rest of this paper is organized as follows. Section II describes the details of wind farm data and key observations on ramps. In Section III, we introduce the proposed model. We verify the model in Section IV and compare it by using realistic wind measurement data from an actual wind farm. This paper concludes in Section V.

II. Data Description and Key Observations

The data used in this paper is from a large wind farm with a rated capacity of 300.5MW, where there are two classes of wind turbines: one class of 53 GE turbines with a rated capacity of 1.5MW and the other class of 221 Mitsubishi turbines with a rated capacity of 1MW. Each class of wind turbines has different power curves as well as cut-in and cut-off speed. For each class, a meteorological tower (MET) is deployed and collocated with a wind turbine. The instantaneous power outputs of both GE turbines $P_{ge}(t)$ and Mitsubishi turbines $P_{mit}(t)$ and the wind speed $W_s(t)$ and direction $W_d(t)$ measured at all the METs are recorded every 10 min for the years of 2009 and 2010. $W_d(t)$ is measured in degree that is different from the previous measurement.

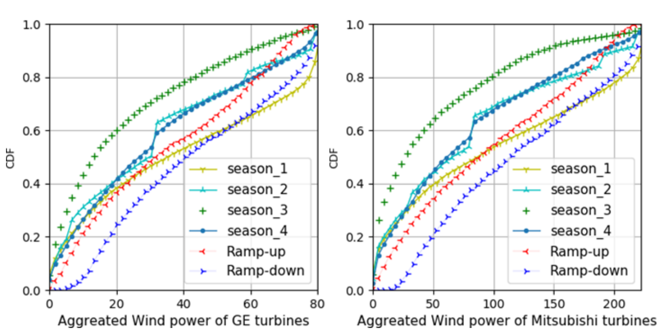

Fig. 1: Empirical distribution of power outputs of GE and Mitsubishi turbines in 4 seasons and ramp events, where season 1 is from January to March.

From the data, we observe 1) the non-stationarity and the seasonality of wind farm power outputs and 2) different dynamics of wind non-ramp, ramp-up, and ramp-down events across different classes of turbines. In Fig. 1, the Cumulative Distribution Functions (CDFs) of power outputs of GE and Mitsubishi turbines in 4 different seasons and ramp-up and ramp-down events are provided, where the ramp-up and ramp-down events are defined if the power output difference in 30 min is greater than 5% and less than 15% of the rated capacity of each class of turbine [17], [18]. It is observed from Fig. 1 that the CDFs of power outputs in different seasons and different ramp events are quite different, indicating the non-stationarity and the seasonality of wind farm power outputs. Moreover, GE and Mitsubishi turbines' power outputs under wind non-ramp, ramp-up and ramp-down events show similar dynamics, but are not the same.

In our previous works [14], [15], we consider the non-stationarity and the seasonality of wind farm power outputs and develop Markov chain based prediction models. In this paper, we take a closer look at the power output dynamics of GE and Mitsubishi turbines under non-ramp, ramp-up, and ramp-down events, in order to achieve a more accurate wind farm power prediction.

III. Seasonal Self-Evolving Neural Networks

Based on the observations in Sec. II, we aim to develop short-term wind farm generation forecast methods that can consider the seasonal power output dynamics of GE and Mitsubishi turbines under non-ramp, ramp-up, and ramp-down events. Motivated by the fact that artificial intelligence (AI) has recently shown a remarkable success in a wide range of fields, we aim to leverage neural networks to capture these different power output dynamics of GE and Mitsubishi turbines. There have been several attempts towards this direction (e.g., ANNs [19] and LSTM [9]; however, these neural network based works rely on a single model and overlook the extreme ramp events, which results in weak performance when ramp events occur. Moreover, when applying neural networks, it is challenging to tune the topology of neural networks, not to mention the hyperparameter tuning.

To address these challenges, we develop seasonal self-evolving neural networks for short-term wind farm power prediction. The idea is to develop different neural networks for different power output dynamics of GE and Mitsubishi turbines under non-ramp, ramp-up, and ramp-down events and enable these neural networks to self-evolve based on the data, in order to account for the non-stationarity as well as reduce the burden of tuning the topology of neural networks.

The design of the proposed seasonal self-evolving neural networks for short-term wind farm power prediction is illustrated in Fig. 2. We first classify the historical data into non-ramp, ramp-up, and ramp-down datasets, where the non-ramp dataset is further split into 4 datasets for 4 seasons. Then, we leverage NeuroEvolution of Augmenting Topologies (NEAT) [16] to train neural networks, which evolves the neural networks using a genetic algorithm to find the best weighting

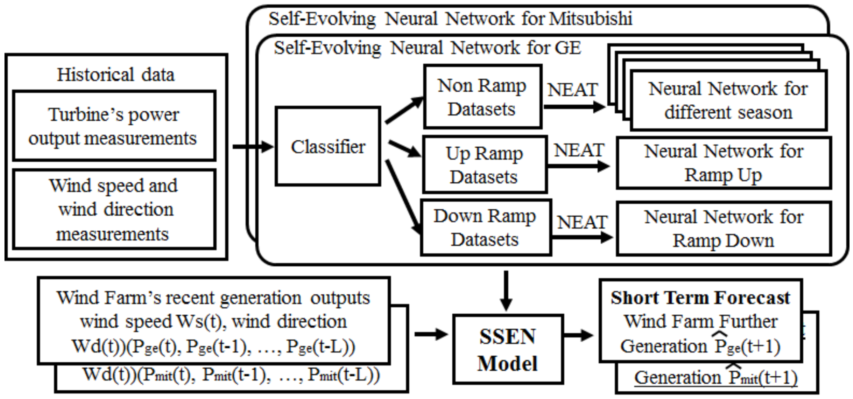

Fig. 2: Illustration of seasonal self-evolving neural networks.

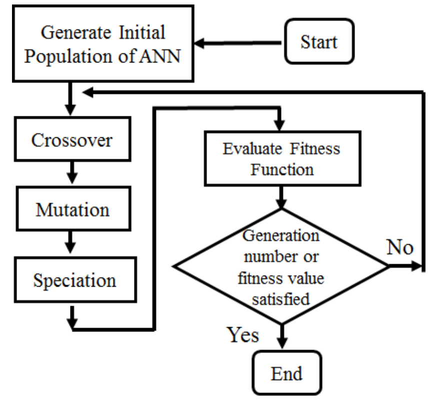

Fig. 3: Workflow of NEAT

parameters and network topology using the classified datasets. Therefore, 12 neural networks (6 for GE and 6 for Mitsubishi) are trained, in order to capture these different power output dynamics of GE and Mitsubishi turbines. Under this design, the model will select appropriate neural networks based on GE and Mitsubishi turbines' recent power outputs to predict short-term wind farm power. The design details of each component are described in the following.

A. Ramp Classifier

The ramp classifier classifies the ramp events based on whether the change of power output in a given time period is larger than a threshold. For example in [17], ramp events are defined if the change of power output between two consecutive hours is greater than or equal to 10% of the nominal capacity of the wind farm.

Specifically, let $Valp_{class}$ and $Valn_{class}$ denote the thresholds associated with ramp up and ramp down events, respectively, where the index $class \in \{GE, Mitsubishi\}$ denotes the type of wind turbines. In a given time window l, the ramp-up and ramp down events can be respectively defined as:

$$P_{class}(t) - P_{class}(t-l) > Valp_{class} \cdot P_{class}^{max}$$
$$P_{class}(t) - P_{class}(t-l) < -Valn_{class} \cdot P_{class}^{max}$$

where $P_{class}(t)$ and P_{class}^{max} denote the power output of one type of wind turbines and maximum value of turbine, respectively.

Using the above definition, the datasets can be classified into up-ramps, down-ramps, and non-ramps for 4 different seasons, i.e., 6 different datasets for each type of wind turbines. Let $X_i^{class}, i = 1, ..., 6$ denote these 6 datasets, where $X_1^{class}, X_2^{class}, X_3^{class}$, and X_4^{class} correspond to the non-ramp datasets for 4 different seasons, X_5^{class} for up-ramp datasets, and X_6^{class} for down-ramp datasets. Clearly, given different values of $Valp_{class}$, $Valn_{class}$, and l, these datasets will change. In this paper, we determine these values based on the prediction accuracy of the corresponding neural networks, as the performance of the neural networks depends on how good the up-ramps, down-ramps, and non-ramps are classified.

B. Self-evolving Neural Networks

For each dataset X_i^{class}, we build a self-evolving neural network to predict wind farm generation. The input of each neural network contains wind speed $W_s(t)$, change of wind direction $W_d(t)$, and recent power outputs $\{P_{class}(t), P_{class}(t-1), ..., P_{class}(t-L)\}$, where L is a parameter determined by the data (see the discussion in Sec. IV-A); the output of the neural network is the predicted power output $\hat{P}_{class}(t+1)$. As illustrated in Fig. 3, we leverage NEAT [16] to train a neural network, which evolves the neural network (NN) using a genetic algorithm (GA) that finds the best weighting parameters and network topology by minimizing the prediction errors, i.e., $\min \sum_t (\hat{P}_{class}(t) - P_{class}(t))^2$.

As illustrated in Fig. 3, NEAT includes random population generation, crossover, mutation, speciation, and evaluation by the fitness function. In this paper, we define the fitness function in terms of forecast accuracy

$$F = -\sum_t (\hat{P}_{class}(t) - P_{class}(t))^2. \quad (1)$$

Each genome in the population represents a NN. The goal is to find the best genome that has the highest fitness (i.e., forecast accuracy). To avoid Competing Conventions Problem [20] or Permutations Problem [21], NEAT uses a direct encoding scheme, in which the neural network architecture is directly encoded into the GA genome [22]. In our case, node gene (list of inputs, hidden nodes, and outputs) and connection gene are used for encoding. Each cell of connection gene specifies the in-node (I), out-node (O), weight of the connection (W), whether enable this connection (E), and the number of innovation (N) that indicates a sequential order of novel generated node. In what follows, we discuss the workflow of NEAT.

We first generate initial population (i.e., a set of genomes) randomly. It is possible that a NN may have no path from its inputs to outputs, and such infeasible genomes will be removed. For the remaining genomes, each genome encodes a NN. Take a NN with 3 inputs ($P_{class}(t), W_s(t), W_d(t)$) and 1 output ($\hat{P}_{class}(t+1)$) as an example. As illustrated in Fig. 4, in the first cell of connect gene, I:1 O:4 W:0.7 denotes connection from Node 1 to Node 4 with weight of 0.7, E:1 represents that it is an enabled connection.

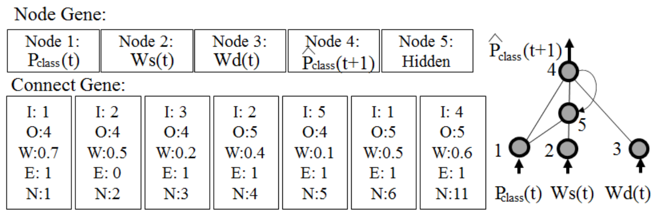

Fig. 4: Encoding of a NN with 3 inputs and 1 output.

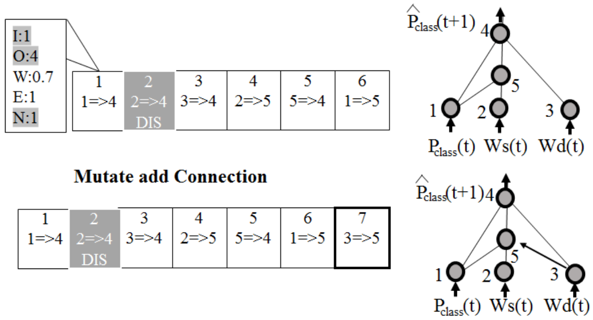

Fig. 5: Mutation by adding a connection to a NN, where a link from Node 3 to Node 5 is added.

In each iteration, each NN can change its connection weights and topology via crossover and mutation, which adds or deletes nodes and connections randomly following the Poisson distribution [16]. Figs. 5 and 6 illustrate possible mutations by adding a connection and a node to a NN, respectively.

After crossover and mutation, topologically similar genomes are classified as the same speciation based on compatibility distance [16]. If the highest fitness of a species does not improve in a predefined generation, the genome in the stagnant species are not allowed to reproduce. After many generations, the species with higher fitness value retains and its elite offspring is saved for prediction.

C. Short-term Wind Farm Generation Forecast

Let the function $H_{\theta_i}^{class}(\cdot)$ denote the NN with parameters θ_i (i.e., the best genome) trained using the dataset X_i^{class}. Given the input $X_i^{class}(t) = \{W_s(t), W_d(t), P_{class}(t), P_{class}(t-1), ..., P_{class}(t-L))\}$, the output of the NN is

$$\hat{P}_{class}(t+1) = H_{\theta_i}^{class}(X_i^{class}(t)). \tag{2}$$

To predict wind farm generation, we need to choose the NN based on the current season and whether ramp events occur or not using the classifier. Using the selected NNs, the wind farm generation forecast $\hat{P}_{ag}(t+1)$ can be obtained by

$$\hat{P}_{ag}(t+1) = \hat{P}_{mit}(t+1) + \hat{P}_{ge}(t+1). \tag{3}$$

Eq. (3) is a point forecast of wind farm generation. To efficiently integrate the wind generation, distributional forecasts are often required to manage the uncertainty [23]. Note that during the training of NNs, we obtain many genomes and we pick the best genomes for the point forecast (3). To obtain the distributional forecast, we can leverage the forecast given by all of these genomes. Let $\{\hat{P}_{ag}^{(j)}(t)\}$ denote the set of forecasts

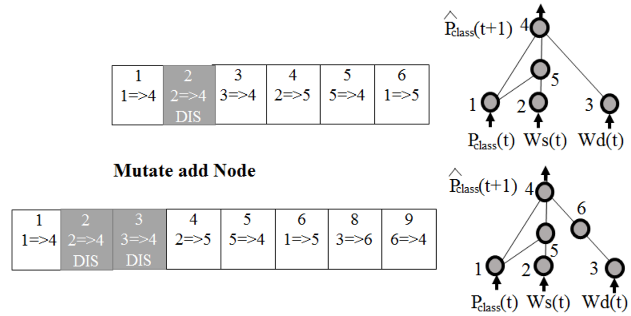

Fig. 6: Mutation by adding a node to a NN, where Node 6 is added between Node 3 and Node 4.

given by each genome j. We assume that the prediction error of the point forecast follows the standard normal distribution. The mean μ_t and the variance σ_t^2 can be estimated as

$$\mu_t = \frac{1}{J}\sum_{j=1}^{J}\hat{P}_{ag}^{(j)}(t) \tag{4}$$

$$\sigma_t^2 = \frac{1}{J}\sum_{j=1}^{J}(\hat{P}_{ag}^{(j)}(t) - \mu_t)^2 \tag{5}$$

where J is the number of genomes. Based on this assumption, we can obtain the $(1-\alpha)$ confidence interval of the point forecast (3) as

$$[\hat{P}_{ag}(t+1) - Z(1-\frac{\alpha}{2})\sigma_{t+1}, \hat{P}_{ag}(t+1) + Z(1-\frac{\alpha}{2})\sigma_{t+1}], \tag{6}$$

where $Z(1-\frac{\alpha}{2})$ denotes the point where the cumulative distribution function of the standard normal distribution equals $1-\frac{\alpha}{2}$.

Remarks. The proposed self-evolving neural networks can be first trained offline and then updated online when new data for each season is available. As each neural network is trained using different datasets, the training can be carried out in parallel. Therefore, the neural networks can be obtained efficiently. Moreover, the training of neural networks does not require AI experts to tune the topology of neural networks and the hyperparameters. Thus, the proposed approach can be easily implemented in practice.

IV. Case Studies of Real Wind Power Data

A. Experimental Setup

1) Data: The description of the data used in the paper can be found in Sec. II. In the case studies, the data of year 2009 is used to train the neural networks, and the data of year 2010 is used to test the forecast accuracy of the proposed method.

2) Performance Measure: We use mean absolute error (MAE) and root-mean square error (RMSE) to quantify the forecast errors, i.e.,

$$MAE = \frac{1}{N}\sum_t \left|\hat{P}_{ag}(t) - P_{ag}(t)\right|$$

$$RMSE = \sqrt{\frac{1}{N}\sum_t \left|\hat{P}_{ag}(t) - P_{ag}(t)\right|^2}$$

where N is the number of data points used for testing.

3) Parameter Tuning: For the proposed seasonal self-evolving neural networks, the parameters $Valp_{class}$, $Valn_{class}$, and L will impact the prediction performance. For $Valp_{class}$ and $Valn_{class}$, we leverage the thresholds (i.e., 10%, 15%, or 20% changes in one hour) provided by the existing works [17], [18] and find the best thresholds for our data via exhaustive search. For L, we pick the best value based on the MAE under different values of L (see Fig. 7). We observe that when $L = 9$, it gives the best MAE. We did not try large L, because 1) the running time increases significantly for large L due to complex evolution process and 2) bad features may be involved due to the non-stationarity of the wind power outputs. During the training phase, the parameters of each NN are selected using the best genome over 20 runs. The obtained genomes will be used for both point and distributional forecasts as described in Sec. III-C.

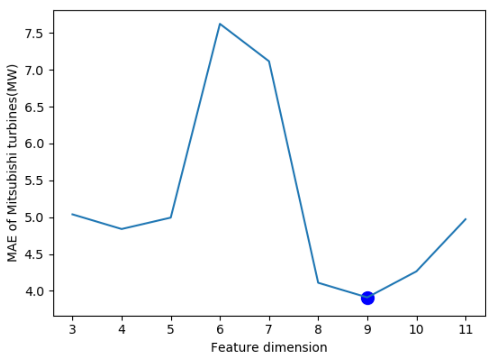

Fig. 7: MAE as the function of dimension.

4) Benchmark: We compare the forecast performance of the proposed seasonal self-evolving neural networks (SSEN) with five other approaches:

- The adaptive AR model [15],
- The Markov-chain-based (MC) model [14],
- The SVM enhanced Markov (SVM-MC) model [15],
- The Seasonal NEAT model without separating ramp and non-ramp events,
- The Long Short-Term Memory (LSTM) model.

The seasonal NEAT model, similar to the proposed approach, considers 4 seasons; however, it does not separate the ramp and non-ramp events during the training phase, which would perform worse for ramp events. For the LSTM model, we use 3 layers, where the first layer is Batch-Normalized and the activation function of the second layer and the third layer are Rectified Linear Unit(ReLu) and tanh, respectively. For each layer, we search for the optimal number of nodes from 20 to 100.

B. Experimental Results

1) 10-min ahead forecast: Tables I and II compare the 10-min ahead forecast performance of different approaches for the entire year 2010 and all ramp events in year 2010, respectively.

TABLE I
COMPARISON OF DIFFERENT FORECAST APPROACHES OVER THE DATA IN 10-MIN AHEAD FORECAST OF THE ENTIRE YEAR 2010

Error	AR	MC	SVM-MC	NEAT	LSTM	SSEN
MAE(%)	2.441	2.413	2.214	1.778	1.799	**1.704**
RMSE(%)	3.974	3.525	3.342	3.074	3.072	**3.023**

TABLE II
COMPARISON OF DIFFERENT FORECAST APPROACHES OVER ALL RAMP EVENTS IN 10-MIN AHEAD FORECAST OF THE ENTIRE YEAR 2010

Error	AR	MC	SVM-MC	NEAT	LSTM	SSEN
MAE(%)	2.945	2.856	2.657	2.416	2.469	**2.320**
RMSE(%)	4.403	3.837	3.655	3.668	3.679	**3.534**

It is observed that the proposed approach outperforms the other approaches. Specifically, the improvement of the proposed approach in terms of MAE is at least 23% in all events and 12% in ramp events over non-NN based approaches (AR, MC, and SVM-MC) and at least 4.1% in all events and 3.9% in ramp events over other NN based approaches (NEAT and LSTM). This is because the ramp classification separates the ramp and non-ramp events and self-evolving neural networks can more effectively learn different power output dynamics of GE and Mitsubishi turbines under non-ramp, ramp-up, and ramp-down events.

Figs. 8, 9, and 10 provide three episodes of prediction intervals. January 14, 2010 is chosen for illustration, in which the wind power fluctuates mildly with an average ramp rate of 6 MW per hour. January 16, 2010 is selected due to remarkable wind power fluctuation from midnight to 00:50 with an average ramp rate of 32 MW per hour. February 7, 2010 is picked since there are ramp-up and ramp-down events in 3 hours with an average ramp rate around 13 MW per hour. As illustrated in these figures, the actual wind farm generation lies in the prediction interval obtained from (6), despite the sharp ramps.

2) 60-min ahead forecast: Tables III and IV compare the 60-min ahead forecast performance of different approaches for the entire year 2010 and all ramp events in year 2010, respectively.

TABLE III
COMPARISON OF DIFFERENT FORECAST APPROACHES OVER THE DATA IN 60-MIN AHEAD FORECAST OF THE ENTIRE YEAR 2010

Error	AR	MC	SVM-MC	NEAT	LSTM	SSEN
MAE(%)	9.624	8.727	8.727	8.277	8.257	**7.862**
RMSE(%)	20.869	13.514	13.514	11.884	12.357	**11.706**

TABLE IV
COMPARISON OF DIFFERENT FORECAST APPROACHES OVER ALL RAMP EVENTS IN 60-MIN AHEAD FORECAST OF THE ENTIRE YEAR 2010

Error	AR	MC	SVM-MC	NEAT	LSTM	SSEN
MAE(%)	11.997	11.091	11.090	10.385	10.914	**9.849**
RMSE(%)	21.124	15.527	15.527	13.749	14.688	**13.336**

It is observed that the proposed approach also outperforms the other approaches when the prediction horizon increases. Specifically, the improvement of the proposed approach in terms of MAE is at least 9.9% in all events and 11.2% in ramp events over non-NN based approaches (AR, MC, and SVM-MC) and at least 4.8% in all events and 5.1% in ramp events over other NN based approaches (NEAT and LSTM).

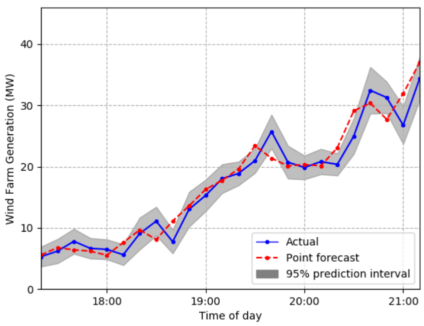

Fig. 8: January 14, 2010

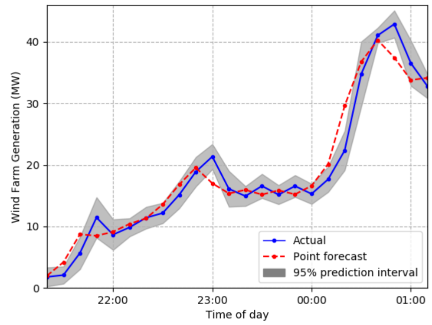

Fig. 9: January 16, 2010

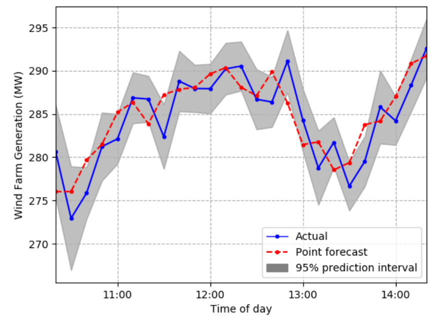

Fig. 10: February 7, 2010

3) Discussions: Based on the experimental results, the performance of NN-based approaches is better than non-NN-based approaches. By splitting the training datasets into multiple training datasets for different dynamics of wind farm generation, the proposed approach can better learn such heterogeneous dynamics and the proposed NNs can evolve with time to track the changing dynamics. Therefore, the proposed approach can outperform the other approaches.

V. Conclusion

In this paper, the seasonal self-evolving neural network model for short-term wind farm generation forecast is developed. This approach takes into consideration the non-stationarity and the seasonality of wind farm generation as well as different dynamics of non-ramp, ramp-up, and ramp-down events across different classes of wind turbines. Specifically, the historical data of the wind turbine power outputs are first classified into ramp-up and ramp-down datasets and non-ramp datasets for different seasons. To capture heterogeneous wind dynamics, different neural networks for each dataset are trained separately. To find the best weighting parameters and network topology, we leverage NEAT to train these neural networks. Based on the proposed self-evolving neural networks, short-term distributional forecasts and point forecasts are then derived. Experimental results demonstrate the significantly improved accuracy of the proposed forecast approach.

Acknowledgment

The authors are grateful to the National Renewable Energy Laboratory (NREL) and Xcel Energy for providing the data of the wind farm used in this study.

References

[1] K. S. Cory and B. G. Swezey, “Renewable portfolio standards in the states: Balancing goals and implementation strategies,” tech. rep., National Renewable Energy Lab (NREL), 2007.

[2] F. Cassola and M. Burlando, “Wind speed and wind energy forecast through kalman filtering of numerical weather prediction model output,” *Applied energy*, vol. 99, pp. 154–166, 2012.

[3] P. Pinson and H. Madsen, “Adaptive modelling and forecasting of offshore wind power fluctuations with markov-switching autoregressive models,” *Journal of forecasting*, vol. 31, no. 4, pp. 281–313, 2012.

[4] G. Papaefthymiou and B. Klockl, “Mcmc for wind power simulation,” *IEEE transactions on energy conversion*, vol. 23, no. 1, pp. 234–240, 2008.

[5] Y. Liu, J. Shi, Y. Yang, and W.-J. Lee, “Short-term wind-power prediction based on wavelet transform–support vector machine and statistic-characteristics analysis,” *IEEE Transactions on Industry Applications*, vol. 48, no. 4, pp. 1136–1141, 2012.

[6] G.-f. Fan, W.-s. Wang, C. Liu, and H.-z. DAI, “Wind power prediction based on artificial neural network,” *Proceedings of the CSEE*, vol. 28, no. 34, pp. 118–123, 2008.

[7] H. Chitsaz, N. Amjady, and H. Zareipour, “Wind power forecast using wavelet neural network trained by improved clonal selection algorithm,” *Energy conversion and Management*, vol. 89, pp. 588–598, 2015.

[8] G. Osório, J. Matias, and J. Catalão, “Short-term wind power forecasting using adaptive neuro-fuzzy inference system combined with evolutionary particle swarm optimization, wavelet transform and mutual information,” *Renewable Energy*, vol. 75, pp. 301–307, 2015.

[9] X. Shi, X. Lei, Q. Huang, S. Huang, K. Ren, and Y. Hu, “Hourly day-ahead wind power prediction using the hybrid model of variational model decomposition and long short-term memory,” *Energies*, vol. 11, no. 11, p. 3227, 2018.

[10] Y. Hao and C. Tian, “A novel two-stage forecasting model based on error factor and ensemble method for multi-step wind power forecasting,” *Applied energy*, vol. 238, pp. 368–383, 2019.

[11] Q. Wu and C. Peng, “Wind power generation forecasting using least squares support vector machine combined with ensemble empirical mode decomposition, principal component analysis and a bat algorithm,” *Energies*, vol. 9, no. 4, p. 261, 2016.

[12] A. A. Abdoos, “A new intelligent method based on combination of vmd and elm for short term wind power forecasting,” *Neurocomputing*, vol. 203, pp. 111–120, 2016.

[13] D. Ding, M. Zhang, X. Pan, M. Yang, and X. He, “Modeling extreme events in time series prediction,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1114–1122, 2019.

[14] M. He, L. Yang, J. Zhang, and V. Vittal, “A spatio-temporal analysis approach for short-term forecast of wind farm generation,” *IEEE Transactions on Power Systems*, vol. 29, no. 4, pp. 1611–1622, 2014.

[15] L. Yang, M. He, J. Zhang, and V. Vittal, “Support-vector-machine-enhanced markov model for short-term wind power forecast,” *IEEE Transactions on Sustainable Energy*, vol. 6, no. 3, pp. 791–799, 2015.

[16] K. O. Stanley and R. Miikkulainen, “Evolving neural networks through augmenting topologies,” *Evolutionary computation*, vol. 10, no. 2, pp. 99–127, 2002.

[17] C. W. Potter, E. Grimit, and B. Nijssen, “Potential benefits of a dedicated probabilistic rapid ramp event forecast tool,” in *2009 IEEE/PES Power Systems Conference and Exposition*, pp. 1–5, IEEE, 2009.

[18] J. Freedman, M. Markus, and R. Penc, “Analysis of west texas wind plant ramp-up and ramp-down events,” *AWS Truewind, LLC*, 2008.

[19] G. Chang, H. Lu, Y. Chang, and Y. Lee, “An improved neural network-based approach for short-term wind speed and power forecast,” *Renewable energy*, vol. 105, pp. 301–311, 2017.

[20] D. J. Montana and L. Davis, “Training feedforward neural networks using genetic algorithms.,” in *IJCAI*, vol. 89, pp. 762–767, 1989.

[21] N. J. Radcliffe, “Genetic set recombination and its application to neural network topology optimisation,” *Neural Computing & Applications*, vol. 1, no. 1, pp. 67–90, 1993.

[22] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.

[23] L. Yang, M. He, V. Vittal, and J. Zhang, “Stochastic optimization-based economic dispatch and interruptible load management with increased wind penetration,” *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 730–739, 2016.