2021 IEEE International Conference on Environment and Electrical Engineering and 2021 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe) | 978-1-6654-3613-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/EEEIC/ICPSEurope51590.2021.9584664

An Advanced Generative Deep Learning Framework for Probabilistic Spatio-temporal Wind Power Forecasting

Seyed Mohammad Jafar Jalali
IISRI
Deakin University
Victoria, Australia
sjalali@deakin.edu.au

Mahdi Khodayar
Department of Computer Science
University of Tulsa
Tulsa, OK, USA
mahdi-khodayar@utulsa.edu

Abbas Khosravi
IISRI
Deakin University
Victoria, Australia
abbas.khosravi@deakin.edu.au

Gerardo J. Osório
C-MAST/UBI
Portucalense University Infante D. Henrique
Covilha, Portugal
gjosilva@gmail.com

Saeid Nahavandi
IISRI
Deakin University
Victoria, Australia
saeid.nahavandi@deakin.edu.au

João P. S. Catalão
INESC TEC
Faculty of Engineering of the
University of Porto (FEUP)
Porto, Portugal
catalao@fe.up.pt

***Abstract*—This paper presents a deep generative model for capturing the conditional probability distribution of future wind power given its history by modeling and pattern recognition in a dynamic graph. The dynamic nodes show the wind sites while the dynamic edges reflect the correlation between the nodes. We propose a scalable optimization model, which is theoretically proved to catch distributions at nodes of the graph, contrary with all learning formulations in the sector of discriminatory pattern recognition. The density of probabilities for each node can be used as samples in our framework. This probabilistic deep convolutional Auto-encoder (PDCA), is based on the deep learning of localized first-order approximation of spectral graph convolutions, a novel evolutionary algorithm and the Bayesian variational inference concepts. The presented generative model is used for the spatio-temporal probabilistic wind power problem in a wide 25 wind sites located in California, the USA for up to 24 hr ahead prediction. The experimental findings reveal that our proposed model outperforms other competitive temporal and spatio-temporal algorithms in terms of reliability, sharpness, and continuous ranked probability score.**

***Index Terms*—Deep Learning, Probabilistic Forecasting , Variational Bayesian Inference, Spectral Graph Convolutions, Evolutionary Algorithm**

I. Introduction

Energy supply scarcities have emerged a severe challenge due to the rapid growth in the magnitude of production and trade consumption [1]. Wind power is considered to become one of the most attractive sectors to address the energy shortage and has been welcomed by its accessible source low pollution and cheaper cost [2]. Within the past few years, the wind power sector plays a significant role throughout the global economy, and the annual growth rate of worldwide wind power has steadily continued to enhance. Nevertheless, the natural fluctuations and generation capacity of wind can induce the non-stationary and dynamic origin of wind power, that further has a detrimental effect on the performance of energy grid [3]. In particular, the potentially diverse and large deployment of wind power consequently possesses significant negative impacts on the transmission of the power system which leads in decreasing the performance of the power system [4]. The forecasting of wind power is considered to be a key strategy to the solving of this issue. Thus the accurate forecasting of wind power is considered to become a key strategy for the wind farms and power generation systems to establish the effective transmission, allocation and installation paradigms for the sustainable performance of the energy grid sectors [3], [5], [6].

Generally, the methodologies for wind power forecasting in the literature are segmented into three main categories including persistence models, physical models as well as statistical and artificial intelligence (AI) algorithms [7]. Furthermore, in the recent years, the deep neural network technologies have raised lots of attention by the researchers who are active in probabilistic forecasting aspect of wind power due to their excellent performance for prediction tasks [8]. The authors in [9] proposed an improved deep mixture density network of multiple wind farms for the short term wind power probabilistic forecasting. To this end, the a deep multi-to-multi neural network model is developed to generate probabilistic forecasts in an end-to-end way, which takes the beta kernel as an element to prevent the difficulties of leakage. The study in [10] outlines

J.P.S. Catalão acknowledges the support by FEDER funds through COMPETE 2020 and by Portuguese funds through FCT, under POCI-01-0145-FEDER-029803 (02/SAICT/2017). G.J. Osório acknowledges the support by UIDB/00151/2020 research unit (C-MAST) funded by FCT.

978-1-6654-3613-7/21/$31.00 ©2021 IEEE

a new efficient approach for probabilistic wind power forecasting relying on deep spiking neural network. For this strategy, an unique predicting methodology is developed to compute the coverage probability and sharpness with associated confidence levels. After this, group search optimizer has been initiated to optimize the parameters of the predicting model and effectively start generating the prediction intervals for maintaining the prediction robustness and reliability.

In this study, to highlight the spatiotmporal wind power pattern recognition in a probabilistic wind forecasting setting, we develop a novel deep generative model that learns the underlying conditional probability distribution function (PDF) of future wind power of multiple sites given their historical data. The proposed approach defines an encoding-decoding neural architecture. The encoder is presented as a graph convolutional neural network that captures the significant spatiotemporal features of the wind data while the decoder applies a feed-forward neural network with Rough neurons to map the captured spatiotemporal wind features to the wind power forecast values. The Rough neurons enhance the reliability of the framework by capturing interval knowledge corresponding to the weights and biases of the encoder as well as the decoder.

In addition, we design a novel efficient evolutionary algorithm based on two enhancement modifications over the competitive swarm optimizer algorithm (ECSO) to efficiently optimize the weights and biases of the proposed NN. This strategy is named as neuroevolution which efficiently optimize the NN architecture [1], [11]–[13].

This paper is organized as follows: In Section II, the roblem formulation is presented and proposed PDCA model is described in details in Section III. Also, the enhanced evolutionary algorithm is presented in Section IV. The experimental results are discussed in Section V and the paper is finally concluded in Section VI.

II. Problem Formulation

We collect spatioemporal WP time series from $M = 25$ site locations in California for the year 2006 by National Solar Radiation Data Base [14]. The locations of the sites are depicted in Fig. 1. The data collected on each site incorporates the 5-min interval WP time series. Let us describe the weighted undirected WP graph, $G = \langle N_G, L_G \rangle$, where N_G is made up of $n_i,\ i \in \{1, 2, ..., 25\}$ nodes. i.e., WP site locations, and L_G is the collection of edges between graph nodes. The present mutual knowledge (MI) and geographical distance within nodes across the graph are used to generate the edge weight matrix, W. More technically, we interpret the i-th and j-th components of weight matrix denoted by W as following:

$$w_{ij} = \begin{cases} e^{-D_{ij}} & MI(i,j) \geq \lambda \\ 0 & MI(i,j) < \lambda \end{cases} \tag{1}$$

D_{ij} repreents the geographical distance among i and j nodes, whereas $MI(i,j)$ standardizes the MI between these two nodes. For this study, we assigned $\lambda = 0.5$ as the sparsity of the edge. We have $S^t_{n_i}$ as a WP time series in the ith node at the time of t. Here, we represent the task of estimating $S^*(t^{'} = t+h)$ for future time series values for a predicted horizon of $h > 0$. In order to predict the mentioned values among all graph points G we have to continue to learn the conditional probability density function (PDF), $P^*\left(S^*(t^{'})|\psi\right)$ in which ψ is the context of all nodes of WP generation. We can further calculate future WP values for each site once we 've learnt P^*.

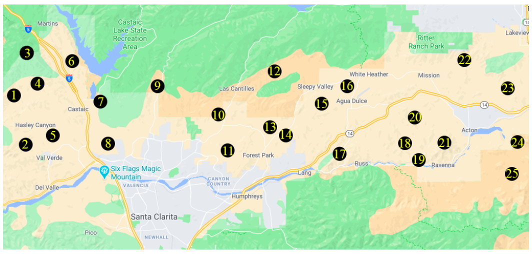

Fig. 1. Locations of 25 underlying WP sites in California, USA.

To select the historical features, we observe that $s(\widetilde{t})$ is more associated towards the most current WP values. As a result, throughout this research, we select historical WP values with $MI > 0.4$ as our input for each wind site in order to forecast the future wind power $s(\widetilde{t+h})$ with forecast horizon $h > 0$.

III. Spatiotemporal Deep Generative Forecasting

A. Graph Convolutional Neural Network

In our proposed model, we generally intend to obtain the PDF of the wind power (WP) to forecast the future values of the time series. Thus, in order to acquire the related spatio-temporal features $(F(G))$ with the graph's nodes, we utilized graph convolutional networks to pattern the WP locations into an undirected weighted graph. The spectral graph convolution for G is determined for time t by:

$$\mathcal{F}_\theta * S^t = V \mathcal{F}_\theta V^T S^t \tag{2}$$

where S^t denotes every node within graph G during time t. The matrix V includes the eigenvector of the normalized Laplacian $L = V \Omega V^t$, and the vector $\theta \in R^n$ is the set of parameters for the frequency domain convolutional filter $\mathcal{F}_\theta = diag(\theta)$. As a function for the L matrix, the $\mathcal{F}_\theta$ feature is considered to be the proprietary vector, the $\mathcal{F}_\theta(\Omega)$ is signified in the filter. By calculating $\mathcal{F}_\theta(\Omega)$ with Polynomials Chebyshev, T_j, we need $\mathcal{F}_\theta(\Omega) \approx \sum_{j=0}^{J} \alpha_j T_j \left(\frac{2}{\vartheta_{max}} \Lambda - I\right)$ for which j-th Chebyshev coefficient of α_j and ϑ_{max} denote to the maximal eigen value for L. The new expression for the Spectral Graph Curvolution Function of G gives us a replacement for this $\mathcal{F}_\alpha(\Omega)$ (2) approximation:

$$\mathcal{F}_\theta * S^t \approx \sum_{j=0}^{J} \alpha_j T_j \left(\frac{2}{\vartheta_{max}} \Lambda - I\right) S^t \tag{3}$$

The convolution formula can be further simplified in (3) with the assumption that it is $J = 1$, $\vartheta_{max} = 2$ and $\alpha_0 = -\alpha_1$. The abbreviated form of (3) is as follows:

$$\mathcal{F}_\theta * S^t \approx \alpha_0 T_0(\Lambda - I)S^t + \alpha_1 T_1(\Lambda - I)S^t = \alpha_0 \left(I + B^{-\frac{1}{2}} W B^{-\frac{1}{2}}\right) S^t \quad (4)$$

The proposed approach shown in Fig. 2 takes LG layers into the Graph feature extractor (GFE) block to extract the spacial-temporal features of the nodes in Graph G through a convolutional (4) process. The k-th level performance of GEF, O^k, is achieved with the following:

$$O^k = f(MO^{k-1}\beta^k) \;\; s.t. \; M = \tilde{B}^{-\frac{1}{2}}\left(W + I\right)\tilde{B}^{-\frac{1}{2}} \quad (5)$$

where W represents the graph weight matrix specified in (1), β^k is the weights of the k-th layer of a neural network (NN), and$\tilde{B}_{ij} = \sum_j \left(W + I\right)_{ij}$. The GEF network input is the raw data of the historical time series and the output of $F(G)$ are spatio-temporal features for graph G.

B. Time Series Approximation by PDF Learning

The aim of this Section is to acquire the $P(X)$ for large dimensional data points $X \in \mathcal{X}$. Afterward, in Section III-B2, we apply the mathematics to our problem, revealing the $P^* \left(S^* | \psi\right)$. With P^*, we can produce future values of time series, S^*, that are as similar to the observed samples, S, while feasible during the training process.

1) Learning Probabilistic Representation of the Data: As the size of the input space $\mathcal{X}$ increases, thus does the great difficulties of accurate $P(X)$ estimation. As a result, we project the inputs through an implicit randomized domain, $\mathcal{Z}$, in order to reflect the most noticeable advantage of $P(X)$. In this case, we can assure that by scraping via an unspecified distribution $P(z)$ within a high-dimensional domain $\mathcal{Z}$, we can obtain certain samples $\hat{X}$ that match the original PDF $P(X)$. Consider a range of deterministic functions $f(z;\theta)$ including parameters $\theta \in \Theta$ for converting data points from $\mathcal{Z}$ to $\mathcal{X}$ domains, for example, $f : \mathcal{Z} \times \Theta \to \mathcal{X}$. Our aim is to explore a selection of optimal parameters $\theta^* \in \Theta$ for f ensuring that whenever $z \sim P(z)$, the possibility of producing samples X^* that are as similar to X as possible by f is greatly increased. As a result, this optimization model is given as follows:

$$\theta^* = arg \underset{\theta}{max} \left\{ P\left(X\right) = \int f\left(z;\theta\right) P\left(z\right) dz \right\} \quad (6)$$

While $\mathcal{Z}$ represents a generalized vector of the input space $\mathcal{X}$, $f(z;\theta)$ is a randomized matrix in the space $\mathcal{X}$ for a given array of parameters θ. As a result, $P(X)$ in (6) can be expressed in the following form

$$P\left(X\right) = \int D\left(X|z;\theta\right) P\left(z\right) dz \quad (7)$$

where D represents the decoder NN in variational autoencoder (VAE) algorithm.

We interpret conditional PDF $D\left(X|z;\theta\right)$ and subsequent PDF $P(z)$ as Gaussian distributions, $\mathcal{N}(X|f(z;\theta), \sigma^2 \times I)$ and $\mathcal{N}(0, I)$, accordingly. The first layer of the decoder network translates parameters $z \in \mathcal{Z}$ through unidentified problematic functions ζ using variational neurons, and then ζ offers samples $X \in \mathcal{X}$. With the stated definitions, then let reform the optimization technique in (6),(7) as follows:

$$\theta^* = arg \underset{\theta}{max} \int \mathcal{N}(X|f(z;\theta), \sigma^2 \times I)\mathcal{N}(0, I)dz \quad (8)$$

We have to settle on the effectiveness of $z \in \mathcal{Z}$ arbitrary when making new observations of $\mathcal{X}$ (8) in order to address the optimization problem. Therefore, we set the $E(z|X)$ distribution of the conditional probability. Kullback–Leibler (KL) divergence is derived from the expectation value $D(X|z)$ w.r.t. z, $\mathbb{E}_{z\sim E}\left[D\left(X|z\right)\right]$:

$$KL\left[E\left(z|X\right), D\left(z|X\right)\right] = \mathbb{E}_{z\sim E}\left[\log E\left(z|X\right) - \log D\left(z|X\right)\right] \quad (9)$$

Using Bayes rules over $D\left(z|X\right)$, Eq.(9) can be rewritten in:

$$\text{KL}[E\left(z|X\right), P\left(z|X\right)] = \mathbb{E}_{z\sim E}\left[\log E\left(z|X\right) - \log\left(\tfrac{P(X|z)P(z)}{P(X)}\right)\right] = \\ \text{E}_{z\sim E}\left[\log E\left(z|X\right) - \log P\left(X|z\right) - \log P\left(z\right) + \log P\left(X\right)\right] \quad (10)$$

where $P(X|z)$ represents a decoder NN; Thus, we represent it using $D(X|z)$. The aforementioned equality is written in the following form:

$$\log P\left(X\right) - KL\left[E\left(z|X\right)\|P\left(z|X\right)\right] = \\ \mathbb{E}_{z\sim E}\left[\log D\left(X|z\right) - KL\left[E\left(z|X\right)\|P\left(z\right)\right]\right] \quad (11)$$

For our purpose, i.e., generating $X^* \approx X$, we have to maximize $\log\left(P\left(X\right)\right)$ and minimize $KL\left[E\left(z|X\right)||P\left(z|X\right)\right]$ on the left side of the (11); Therefore, we maximize the right side of (11) by stochastic gradient descent (SGD) approach. We can technically describe this optimization problem as follows:

$$\theta^* = arg \underset{\theta}{max} \; \mathbb{E}_{X\sim\mathcal{X}} \begin{bmatrix} \mathbb{E}_{z\sim E}\left[\log D\left(X|z;\theta\right)\right] \\ -KL\left[E\left(z|X;\theta\right)\|P\left(z;\right)\right] \end{bmatrix} \quad (12)$$

Notice that E is a NN encoder to encode input samples X in z (11), and D is a NN encoder from z to X. We adjust E by:

$$E(z|X) = \mathcal{N}(z|\mu(X;\theta), \Sigma(X;\theta)) \quad (13)$$

where μ and Σ are the deterministic functions obtained by a NN with tuanable parameters θ. Since D and E both are multivariate Guassian distribution, the KL term in (11) can be simplified by:
Where the deterministic functions of μ and Σ are NNs by the tuanable parameters θ. The KL expression (11) can be simplified as follows because the D and US are based on multivariate Guassian distribution:

$$\text{KL}[E\left(z|X\right)\|P\left(z\right)] = KL\left[\mathcal{N}(z|\mu(X;\theta), \Sigma(X;\theta)), \mathcal{N}\left(0, I\right)\right] \\ = \frac{1}{2}\left[-\log\left(det\left(\Sigma\right)\right) - d + tr\left(\Sigma\right) + \mu^T\mu\right] \quad (14)$$

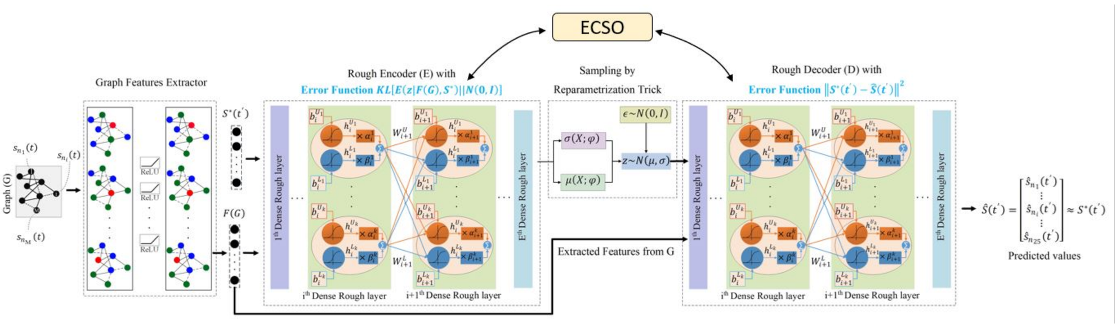

Fig. 2. Proposed Deep Generative Model for Probabilistic Spatiotemporal Forecasting

The trick fpr z Using reparametrization for obtaining $D\left(X|z;\theta\right)$ can be replaced as following:

$$\theta^* = \underset{\theta}{arg\ max}\ \mathbb{E}_{X\sim\mathcal{X}}\left[\begin{matrix}\mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}\left[\begin{matrix}\log[D(X|z=\mu(X)\\ +\Sigma^{1/2}(X)*\epsilon;\theta)]\end{matrix}\right]\\ -KL\left[E\left(z|X;\theta\right)\|P\left(z\right)\right]\end{matrix}\right] \tag{15}$$

The NN encoder receives input and outcome data from μ and Σ (refer to 13) during training procedure. E NN error is calculated in (14). When μ and Σ have been obtained, we can achieve z and generate D for the decoder's network to produce $X^* \approx X$. Notice D NN's error function is $||X - X^*||^2$.

2) Convolutional Graph Rough VAE: This section extends the model for $P^*(S^*|\psi)$ learning procedure explained in III-B1. Our aim is to derive $\hat{S}(t') \approx S^*$, therefore let us formulate the computation phases of S^* inside an estimation of S^*. Thus, the $z \sim E$ as the Bayes rule is assumed to be applied $\log[P(S^*(t')|z,\psi)]$ by:

$$\mathbb{E}_{z\sim E}[\log P(S^*(t')|z,\psi)] = \mathbb{E}_{z\sim E}[\log P(z|S^*(t'),\psi) - \log P(z|\psi) + \log P(V^*(t'|\psi))] \tag{16}$$

(16) can be rewritten as:

$$\log P\left(S^*(t')|\psi\right) - KL\left[E\left(z|S^*(t'),\psi\right)\|P\left(z|S^*(t'),\psi\right)\right] = \mathrm{E}_{z\sim E}\left[\log D\left(S^*(t')|z,\psi\right) - KL\left[E\left(z|S^*(t'),\psi\right)\|P\left(z|\psi\right)\right]\right] \tag{17}$$

As the same as (11), our goal is to make maximum of the left side of (17). Thus, we learn the optimum Rough encoder/decoder in order to grab the conditional PDF $P^*(S^*|\psi)$ through addressing similar optimization problem in (12). With this ideal encoder/decoder, we can create the precise S^* values for future t' values. It should be mentioned that the cost function per adjusting the encoder and the decoder framework is similar to (14).

$$\begin{aligned} Error_E &= KL[E(z|\langle F(G),S^*\rangle\|\mathcal{N}(0,1))] \\ Error_D &= \|\hat{S}(t') - S^*(t')\| \end{aligned} \tag{18}$$

Thus the overall error for the proposed model is $Error_{total} = Error_E + Error_D$.

As mentioned before, in our proposed model, the Rough encoder/decoder NNs are implemented. The rough neurons of the framework contain dynamical interval weight and biases activation functions. The encoder and decoder networks are L_D and L_E Rough layers, respectively. The feedforward on the used Rough network between two layers i and $i+1$ can be also written here by:

$$\begin{aligned} Z_i^U &= W_{i+1}^U O_i + b_{i+1}^U \\ Z_i^L &= W_{i+1}^L O_i + b_{i+1}^L \\ O_{i+1} &= \alpha_{i+1} Z_i^U + \beta_{i+1} Z_i^L \end{aligned} \tag{19}$$

where the upper bound and lower bound (interval weights) of the Rough NNs are respectively denoted by $\left\langle W_{i+1}^U, b_{i+1}^U\right\rangle$ and $\left\langle W_{i+1}^L, b_{i+1}^L\right\rangle$. In encoder/decoder models, our primary reason for using the rough set theory and rough neurons is to increase the strength of the framework over established uncertainties throughout the WP datasets.

IV. The proposed enhanced evolutionary algorithm

In this section, we design an advanced evolutionary algorithm to tune the weights and biases of the NN. This novel approach is named as enhanced competition swarm optimizer (ECSO) in which the basic version of CSO algorithm [15] has been improved using two optimization perorations including levy flight and chaotic map.

First, we begin introducing the CSO algorithm. In CSO, per each iteration, based on the particle positions, two particles are assigned randomly when the particles using the lower fitness value are upgraded towards a better fitness value. The CSO regularly updates the particles using the given equations mainly at t-iteration:

$$v_u^{t+1} = Q_1^t v_u^t + Q_2^t\left(x_w^t - x_u^t\right) + \lambda Q_3^t\left(\bar{x}^t - x_u^t\right) \tag{20}$$

$$x_u^{t+1} = x_u^t + v_u^{t+1} \tag{21}$$

where Q_1^t, Q_2^t, Q_3^t represent randomly the uniform distributed vectors within the range $[0,1]$, x_w^t and x_u^t, respectively denote to the winner and loser particles and the iteration number is represented by t. In t generation, the swarm position average is represented as $\bar{x}^t$, while the λ parameter covers $\bar{x}^t$ particle effectiveness.

For CSO, A is needed to control search agents' step scale that appears to be sequentially reduced with iterations. We employ the strong features of the levy flight strategy in this modifying step to adjust the a parameter. This adjustment continually increases the efficient exploration and exploitation of CSO algorithm. Consider the parameter ∞ uses the following step size:

$$\infty \oplus \text{Levy}(\beta) \sim 0.01 \frac{p}{|q|^{1/\beta}} \left(X_i^k - X_{\text{best}}^k\right) \tag{22}$$

where the values of p and q are defined by:

$$p \sim N\left(0, \phi_u^2\right), q \sim N\left(0, \phi_v^2\right) \tag{23}$$

$$\phi_u = \left[\frac{\Gamma(1+\beta) \times \sin(\pi \times \beta/2)}{\Gamma[(1+\beta)/2] \times \beta}\right]^{1/\beta}, \phi_v = 1 \tag{24}$$

where Γ represents the default gamma function at $[0, 2]$ range. The A parameter with Eq. (25) is amended as follows:

$$\mathbf{A} = Levy(X) * \mathbf{u} \tag{25}$$

where X represents the position of wolves and u is a random value between [0,1] range. These concepts are used to improve the global exploration as well as local exploitation capacity of conventional technology and to deepen the searching advantages of CSO.

The consistency of the preliminary population will also greatly affect the convergence speed and accuracy of the solution by evolutionary algorithms, which are continuously optimized by means of population iteration. The basic CSO often randomly initializes the population and makes it difficult to promise the diversity of the population, due to poor performance efficiency. Consequently, it is vital to increase the diversity of the preliminary population in order to improve the efficiency of the CSO. By applying chaos-theory, the algorithm to local optimal solutions can ideally be eliminated while addressing function optimization problems to maintain the diversity of the population and improve global search effectiveness. There are several strong chaotic maps in literature for various function optimization. In this work, we utilize the great power of Circle map as a one-dimensional chaotic strategy which is an element of probability theory and described by Andrey Colmogorov. For CSO population initialization, we use this map according to the following formula:

$$x_{i+1} = x_k + b - \left(\frac{P}{2\pi}\right) \sin(2\pi x_i) mod(1) \tag{26}$$

In the above formula, chaotic numbers are generated between [0,1] interval with the default configuration as P=0.5 and b=0.2. Both P and b denote to the control parameters.

V. Experimental Results

In order to evaluate the proposed PDCA, we comapre the model with Extreme Learning Machine (ELM), Kernel Density Estimation (KDE), Quantile Regression (QR), Persistence Prediction (PP), Spatiotemporal Copula (ST-Copula), Spatiotemporal-QR-Lasso (ST-QR-Lasso), Spatiotemporal Support Vector Regression (ST-SVR), and Conditional Spatiotemporal Forecast (CSTF). Fig. 3 indicates the average reliability metrics for all wind power sites with different nominal coverage rates between 10 and 90 percent. As can be seen from this figure, the proposed PDCA results in the least average deviation in comparison with other spatio-temporal and temporal algorithms, which shows the superior reliability of this method. Fig. 4 reflects the average reliability of various temporal and spatio-time benchmarks for different lookahead periods which shows that the reliability benefit of PDCA grows with the extension of the forecast time horizon.

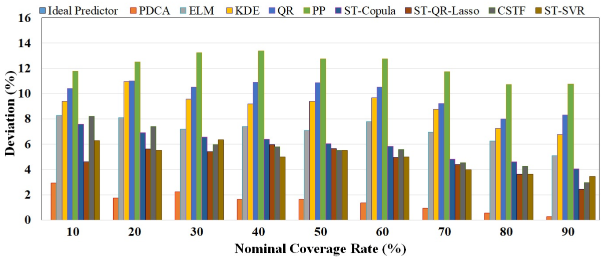

Fig. 3. Reliability measurements averaged over all wind farms

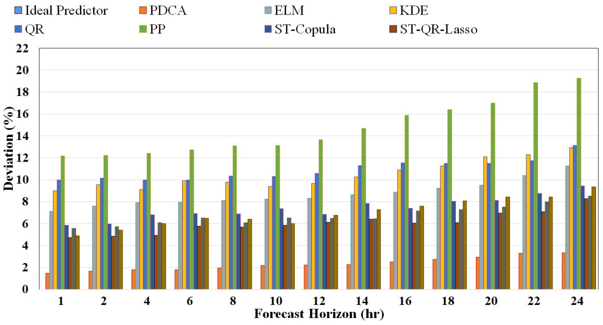

Fig. 4. Average reliability with different look-ahead times

Fig. 5 indicates the 10% -90% average nominal cov-

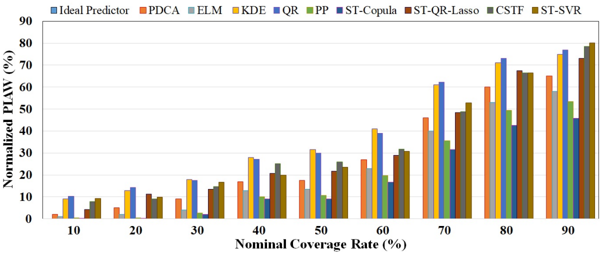

Fig. 5. Sharpness evaluation using normalized PIAW

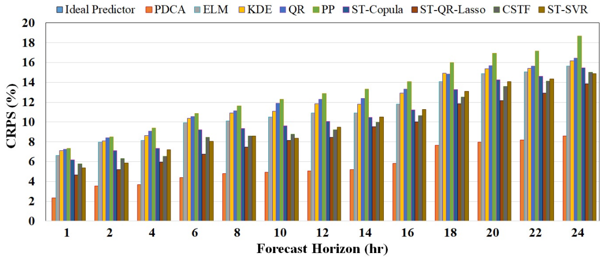

Fig. 6. CRPS outcomes from one-hour to 24-hr ahead forecasting

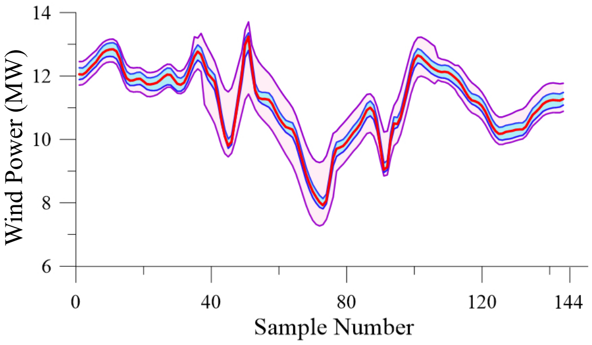

Fig. 7. Probabilistic forecasts and the actual value (red) of 144 wind power samples in March 4th 2006. Blue shows 50% and purple shows 90% confidence intervals of our probabilistic forecast for 24-hr ahead predictions.

erage rates normalizing with the highest observed wind power data points. Among all the models, PCDA offers medium sharpness that is not too significant to guide to wrongly limited quantiles, yet not too small to start losing information regarding potential wind power data. Fig. 6 shows the average Continuous Ranked Probability Score (CRPS) for all wind power predictions from one-hour to 24 hours. The smaller the CRPS of the model, the greater the accuracy it offers. As seen in this diagram, the PDCA model outperforms all other benchmark algorithms. Fig. 7 shows the probabilistic forecasts and the actual value (red) of 144 wind power samples in March 4th 2006. Blue shows 50% and purple shows 90% confidence intervals of our probabilistic forecast for the 24-hr ahead predictions. This figure reveals that real wind power values are followed excellently via our proposed model offering towards highest accuracy, sharpness and reliability.

In overall, compared to previous benchmarks in the literature, experimental findings indicate greater reliability, sharpness and Continuous Ranked Probability Score are obtained by our proposed PCDA framework.

VI. Conclusion

This paper addressed the problem of spatiotmeporal wind power forecasting via deep generative modeling. The probabilistic deep convolutional Auto-encoder, a novel deep generative model for graph-structured datasets, is introduced towards nodal distribution learning problem within dynamic graphs. The nodes reflect the wind power at neighboring wind sites while the edges show their corresponding correlations. The proposed framework determines the corresponding probability densities of nodes by capturing deep convolution features from the underlying graph-structured data. The spatial features of the wind power data are derived via graph spectral convolutions, which are then employed by an encoding and decoding ANN optimized by a novel efficient evolutionary algorithm to capture the distribution of future wind power data points. When compared to recently benchmark methods in the scientific literature, experimental findings represent significant improvement in terms of reliability, sharpness, and continuous ranked probability score.

References

[1] S. M. J. Jalali, S. Ahmadian, A. Khosravi, M. Shafie-khah, S. Nahavandi, and J. P. Catalao, "A novel evolutionary-based deep convolutional neural network model for intelligent load forecasting," *IEEE Transactions on Industrial Informatics*, 2021.

[2] Q. Zhang, H. Zhang, Y. Yan, J. Yan, J. He, Z. Li, W. Shang, and Y. Liang, "Sustainable and clean oilfield development: How access to wind power can make offshore platforms more sustainable with production stability," *Journal of Cleaner Production*, vol. 294, p. 126225, 2021.

[3] S. M. J. Jalali, S. Ahmadian, M. Khodayar, A. Khosravi, V. Ghasemi, M. Shafie-khah, S. Nahavandi, and J. P. Catalão, "Towards novel deep neuroevolution models: chaotic levy grasshopper optimization for short-term wind speed forecasting," *Engineering with Computers*, pp. 1–25, 2021.

[4] T. Simla and W. Stanek, "Influence of the wind energy sector on thermal power plants in the polish energy system," *Renewable Energy*, vol. 161, pp. 928–938, 2020.

[5] H. Hu, L. Wang, and S.-X. Lv, "Forecasting energy consumption and wind power generation using deep echo state network," *Renewable Energy*, vol. 154, pp. 598–613, 2020.

[6] J. Duan, P. Wang, W. Ma, X. Tian, S. Fang, Y. Cheng, Y. Chang, and H. Liu, "Short-term wind power forecasting using the hybrid model of improved variational mode decomposition and correntropy long short-term memory neural network," *Energy*, vol. 214, p. 118980, 2021.

[7] H. Quan, A. Khosravi, D. Yang, and D. Srinivasan, "A survey of computational intelligence techniques for wind power uncertainty quantification in smart grids," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 11, pp. 4582–4599, 2019.

[8] S. M. J. Jalali, P. M. Kebria, A. Khosravi, K. Saleh, D. Nahavandi, and S. Nahavandi, "Optimal autonomous driving through deep imitation learning and neuroevolution," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE, 2019, pp. 1215–1220.

[9] H. Zhang, Y. Liu, J. Yan, S. Han, L. Li, and Q. Long, "Improved deep mixture density network for regional wind power probabilistic forecasting," *IEEE Transactions on Power Systems*, vol. 35, no. 4, pp. 2549–2560, 2020.

[10] H. Wang, W. Xue, Y. Liu, J. Peng, and H. Jiang, "Probabilistic wind power forecasting based on spiking neural network," *Energy*, vol. 196, p. 117072, 2020.

[11] S. M. J. Jalali, S. Ahmadian, A. Kavousi-Fard, A. Khosravi, and S. Nahavandi, "Automated deep cnn-lstm architecture design for solar irradiance forecasting," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021.

[12] S. M. J. Jalali, M. Ahmadian, S. Ahmadian, A. Khosravi, M. Alazab, and S. Nahavandi, "An oppositional-cauchy based gsk evolutionary algorithm with a novel deep ensemble reinforcement learning strategy for covid-19 diagnosis," *Applied Soft Computing*, p. 107675, 2021.

[13] S. J. Mousavirad, S. M. J. Jalali, S. Ahmadian, A. Khosravi, G. Schaefer, and S. Nahavandi, "Neural network training using a biogeography-based learning strategy," in *International Conference on Neural Information Processing*. Springer, 2020, pp. 147–155.

[14] M. Sengupta, Y. Xie, A. Lopez, A. Habte, G. Maclaurin, and J. Shelby, "The national solar radiation data base (nsrdb)," *Renewable and Sustainable Energy Reviews*, vol. 89, pp. 51–60, 2018.

[15] R. Cheng and Y. Jin, "A competitive swarm optimizer for large scale optimization," *IEEE transactions on cybernetics*, vol. 45, no. 2, pp. 191–204, 2014.