Convolutional Graph Autoencoder: A Generative Deep Neural Network for Probabilistic Spatio-temporal Solar Irradiance Forecasting

Mahdi Khodayar, *Student Member, IEEE*, Saeed Mohammadi, *Student Member, IEEE,* Mohammad E. Khodayar, *Senior Member, IEEE* Jianhui Wang, *Senior Member, IEEE,* and Guangyi Liu, *Senior Member, IEEE*

***Abstract*—Machine Learning on graph-structured data is an important and omnipresent task for a vast variety of applications including anomaly detection and dynamic network analysis. In this paper, a deep generative model is introduced to capture continuous probability densities corresponding to the nodes of an arbitrary graph. In contrast to all learning formulations in the area of discriminative pattern recognition, we propose a scalable generative optimization/algorithm theoretically proved to capture distributions at the nodes of a graph. Our model is able to generate samples from the probability densities learned at each node. This probabilistic data generation model, i.e. convolutional graph autoencoder (CGAE), is devised based on the localized first-order approximation of spectral graph convolutions, deep learning, and the variational Bayesian inference. We apply our CGAE to a new problem, the spatio-temporal probabilistic solar irradiance prediction. Multiple solar radiation measurement sites in a wide area in northern states of the US are modeled as an undirected graph. Using our proposed model, the distribution of future irradiance given historical radiation observations is estimated for every site/node. Numerical results on the National Solar Radiation Database show state-of-the-art performance for probabilistic radiation prediction on geographically distributed irradiance data in terms of reliability, sharpness, and continuous ranked probability score.**

***Index Terms*—Graph-structured Data, Deep Generative Model, Spatio-temporal Regression, Probabilistic Forecasting, Spectral Graph Convolutions, Variational Bayesian Inference**

I. INTRODUCTION

In recent years, the rapid exhaustion of fossil fuel sources, the environmental pollution concerns, and the aging of the developed power plants are considered as crucial global concerns. As a consequence, the renewable energy resources including wind and solar have been rapidly integrated into the existing power grids. The reliability of power systems depends on the capability of handling expected and unexpected changes and disturbances in the production and consumption, while maintaining quality and continuity of service. The variability and stochastic behavior of photovoltaic (PV) power caused by the solar radiation uncertainty lead to major challenges including voltage fluctuations, as well as local power quality and stability issues [1], [2], [3]. Hence, accurate solar irradiance forecasting for PV estimation is required for effective operation of power grids [4]. The studies in the area of solar irradiance and PV power forecasting are mainly categorized into four major classes:

1) The persistence model is applied as a baseline that assumes the irradiance values at future time steps is equal to the same values at the forecasting time. Due to such a strong smoothness assumption, the persistence scheme is only effective for intra-hour applications [2].

2) Physical models employ physical processes to estimate the future solar radiation values using astronomical relationships [5], meteorological parameters, and numerical weather predictions (NWPs) [6]. In [7], an hourly-averaged day-ahead PV forecasting approach is presented based on least squares optimization of NWPs using global horizontal irradiance (GHI) and the zenith angle. Some NWPs make use of the clear sky radiation modeled by earth-sun geometry [8] or panel tilt/orientation along with several meteorological parameters such as temperature or wind speed [9]. Other works apply cloud motion vector (CMV) frameworks [10] for accurate short-term predictions, using static cloud images [11], satellite images [12], or the sensor networks [13].

3) Statistical and Artificial intelligence (AI) techniques are recently presented for a number of solar irradiance and PV power estimation/regression problems. As discussed in [14], the non-stationary and highly nonlinear characteristics of solar radiation time series lead to the superiority of AI approaches over the traditional statistical models. Machine learning algorithms are employed as target function approximators, to estimate future solar irradiance or PV power. Highly nonlinear regression methodologies including ANNs [15], [16] and support vector machines/regression (SVM/R) [17] have been employed for short-term purposes. [17] presents a benchmarking of supervised neural networks, Gaussian processes and support vector machines for GHI predictions. In [18], [19] a bootstrapping approach is presented to estimate uncertainties involved in the prediction of wind/solar time series. Here, a number of Extreme Learning Machine (ELM) ANNs are trained as regression models using resampled training data. The uncertainties in solar/wind data and the model uncertainties are modeled as two classes of uncertainties to provide probabilistic

Mahdi Khodayar, Saeed Mohammadi, Mohammad E. Khodayar and Jianhui Wang are with the Department of Electrical Engineering of Southern Methodist University, Dallas, TX, USA (mahdik@smu.edu, smohammadi@smu.edu, mkhodayar@smu.edu and jianhui@smu.edu), Guangyi Liu is with Global Energy Interconnection Research Institute North America (GEIRI North America or GEIRINA), San Jose, CA, USA (email: guangyi.liu@geirina.net)

This work is partially supported by grant ECCS-1710923 National Science Foundation, and State Grid Corporation technology project 5455HJ180018.

predictions. This model has low generalization capability as both uncertainties are associated with a strong prior knowledge that forces the uncertainties to be Gaussian. [20] employs k-nearest neighborhood (k-NN) method to find days with similar weather condition. Kernel Density Estimation (KDE) is further applied to estimate the probability density function (PDF) of PV for the neighbors of k-NN. [21] provides a comprehensive review of non-parametric methods that employ k-NN to find the expected value of their assumed probability distribution functions for solar irradiance and PV forecasting. [22] applies k-NN for short-term predictions with less than 20-min ahead horizons. Also, [23] employs k-NN and gradient boosting with various meteorological measurements such as surface pressure, total cloud cover, and relative humidity for 24-hr ahead forecasts.

Quantile Regression (QR) is another statistical method employed in non-parametric prediction models. In recent literature, QR is well-studied for the estimation of statistical parameters (e.g. mean and variance) of predefined probability distributions for future solar values [21]. In [24], the ELM neural network utilizes a QR-based parameter estimation for hourly solar predictions. Also, [25] employs the combination of QR and ELM for very short-term applications with 5-min horizon length. In [26], a probabilistic prediction model is proposed based on linear QR, combining the point prediction obtained by a deterministic forecasting approach with the information retrieved from ground measurements. Moreover, QR is recently utilized as a non-parametric model in combination with physical methods [21]. In [27], a combination of QR and NWP is presented for daily predictions. Furthermore, [28] proposes an intra-day prediction approach based on multiple QR in combination with the radial basis functions and the alternating direction method of multipliers.

As discussed in [29], [30], fuzzy logic has been recently applied to capture the uncertainties exits in solar datasets. In [31], fuzzy systems are incorporated with neural networks to accurately estimate the real values of future solar irradiance under different sky and temperature conditions. Moreover, [32] presents a fuzzy clustering algorithm to find days with similar irradiance patterns. The solar data corresponding to similar days is further fed to an ELM optimized by Genetic Algorithm (GA) in order to compute daily irradiance predictions. Evolutionary algorithms including GA, Ant Colony [21], and Particle Swarm Optimization [29] help fuzzy systems and ELM to find near-optimal solutions by avoiding erroneous parameter settings caused by poor local optima solutions.

Bayesian approaches have been widely applied to solar prediction problems. In [33], two advanced probabilistic models are proposed based on Bayesian inference for short-term PV prediction. Moreover, new probabilistic indices are presented to compare probabilistic approaches in such a way that the estimated PV values are partially anticipated by the forecasters in their quality-assessment procedures. [34] presents a Naïve Bayes model for the prediction of daily PV energy production. The model uses daily average temperature, total sunshine duration, as well as total global solar radiation to predict future power generation. Furthermore, [35] presents a multi-ahead prediction Multi-Layer Perceptron Neural Network, whose parameters are estimated by a probabilistic Bayesian learning technique. The Bayesian model computes the confidence intervals and estimates the error bars of the Neural Network predictions.

4) Ensemble methods aggregate a set of predictors (i.e. base learners) to increase the prediction accuracy of individual prediction models. As shown by [36], several top-entry PV forecasting models employ ensemble frameworks including QR Forest (QRF) with Gradient Boosting Decision Trees [37], Multiple QR [38], and Gradient Boosting Machines incorporated with NWP [39]. The ensemble models generally use bagging techniques that apply bootstrap sampling to obtain data subsets for training the base learners [38]. Also, some ensemble approaches apply the boosting algorithm which improves the performance of base models by combining them together using a particular cost function (i.e. majority vote) [37], [39]. These techniques decrease prediction variance; hence, prevents the prediction model from overfitting on the training set. In this line of research, [40] proposed a novel probabilistic prediction model based on a competitive ensemble of various base predictors for short-term forecasting of PV power. Three probabilistic methods including Bayesian model, Markov Chain model, and QR were trained as base predictors in order to obtain an ensemble of the predictive distribution with optimal sharpness and reliability metrics. The simulation results of ensemble models show improvement in these metrics compared to single-model methodologies; however, such models need more computational power and increase the time complexity of the predictor [21].

In this paper, a new problem, probability distribution learning in graph-structured data, is solved as a recent pattern recognition challenge. First, generative modeling (learning mathematical patterns from a dataset for the aim of generating new samples under the observed data distribution) is introduced as an optimization problem where the probability of observed data in a given dataset is maximized. Then, our novel graph learning model, Convolutional Graph Autoencoder (CGAE), is presented that is mathematically proved to learn continuous probability density functions from the nodes in an arbitrary graph. Our CGAE is defined based on the first-order approximation of graph convolutions (for learning a compact representation from an input graph) and standard function approximation (more specifically, deep neural architectures with high generalization capacity). The proposed deep learning model is able to generate new samples corresponding to each node, after observing historical graph-structured data, while learning the nodal distributions.

In this study, the problem of spatio-temporal probabilistic solar radiation forecasting is presented as a graph distribution learning problem solved by the CGAE. First, a set of solar measurement sites in a wide area is modeled as an undirected graph, where each node represents a site and each edge reflects the correlation between historical solar data of its corresponding nodes/sites. CGAE is applied to the graph in order to learn the distributions corresponding to the solar data at each site/node. Our CGAE is mathematically guaranteed to efficiently generate samples corresponding to the future solar irradiance values. The samples generated by this model result

in a probabilistic solar radiation forecast for the future time step.

The key contributions of this work are: 1) Our CGAE is the first model devised in the area of machine learning, for the problem of nodal distribution learning in graph-structured data. The presented work is a universal model/algorithm that can be applied to any arbitrary graph for the probability approximation problems. 2) This is the first study of generative modeling for the prediction of renewable resources. Although generative adversarial networks have been applied in [41] for the problem of scenario generation of renewable energy production, this category of machine learning models has not been studied for the prediction tasks as these models do not estimate the probability densities of future observations given the historical measurements. The previous prediction works including all ANNs [15], [18], [19], regression [21], and kernel methods such as SVMs and SVRs [17], as well as all KNN-based methodologies [20], follow discriminative modeling [42], and no generative modeling was introduced in the literature of solar forecasting. Also, in similar areas such as probabilistic load forecasting, most approaches including ANNs [29] and Quantile Regression models [43] are discriminative rather than generative. As shown by the mathematical proof, our generative model leads to accurately understanding the underlying distribution of solar data, while discriminative modeling cannot provide such capability. 3) A spatio-temporal probabilistic forecasting framework is presented that makes use of the knowledge obtained from neighboring solar sites to enhance the prediction reliability and sharpness. 4) In contrast to previous ANN-based approaches [15], [18], [19] that merely apply shallow architectures, i.e. models with a small number of hidden layers, here, our model is able to have as many latent layers as it needs in order to provide the optimal generalization capability to increase the validation accuracy. As a result, the generalization capability and the learning capacity of our proposed deep network are much higher than previous works. Increasing the number of layers in previous models, even with the existence of a regularization error term, is infeasible as it would lead to the vanishing gradient problem. However, here, we solve the issue of having low gradient magnitude that arises in ANN architectures. 5) CGAE is compared with state-of-the-art temporal approaches including Quantile Regression [26], Kernel Density Estimation [20], and Extreme Learning Machine [18], [19] in terms of reliability, sharpness, and Continuous Ranked Probability Score (CRPS) using the National Solar Radiation Database (NSRDB) [44]. Moreover, CGAE is compared with recently proposed state-of-the-art spatio-temporal models including Space-time Copula (ST-Copula) [45], Spatio-temporal QR-Lasso (ST-QR-Lasso) [46], Compressive Spatio-temporal Forecasting (CSTF) [47], and Spatio-temporal Support Vector Regression (ST-SVR) [17], [48]. As shown by the simulation results, CGAE outperforms all temporal as well as spatio-temporal methodologies for 0.5-hr up to 6-hr ahead predictions. CGAE improves the average reliability of the best temporal benchmark, ELM, by 3.64% in hourly predictions which grows to 4.49% in 6-hr ahead forecasting. Moreover, CGAE improves the CRPS of ELM by 3.35% for hourly predictions which is further increased to 5.22% in 6-hr ahead forecasts. Among spatio-temporal approaches, CGAE outperforms all approaches by improving the best spatio-temporal benchmark, ST-SVR, by 2.46% in hourly predictions which is further increased to 4.35% for 6-hr ahead forecasts. CGAE also improves the CRPS of ST-SVR by 1.12% and 4.19% for hourly and 6-hr ahead predictions, respectively. Furthermore, the average widths, as well as the entropies of CGAE's prediction intervals show the significant improvement of prediction sharpness of the proposed method compared to the state-of-the-art benchmarks.

The paper is organized as follows: In Section II the problem of probabilistic solar irradiance forecasting is defined. In section III, first, our proposed generative modeling paradigm is defined mathematically. Then, our CGAE model is formulated and its application for solving the forecasting problem is explained. Theoretical guarantee of the proposed methodology is available in this section. Section IV explains the performance metrics and shows numerical results on a large dataset. Finally, the conclusions and future works on generative modeling are presented in Section V.

II. Problem Formulation for Probabilistic Solar Irradiance Forecasting

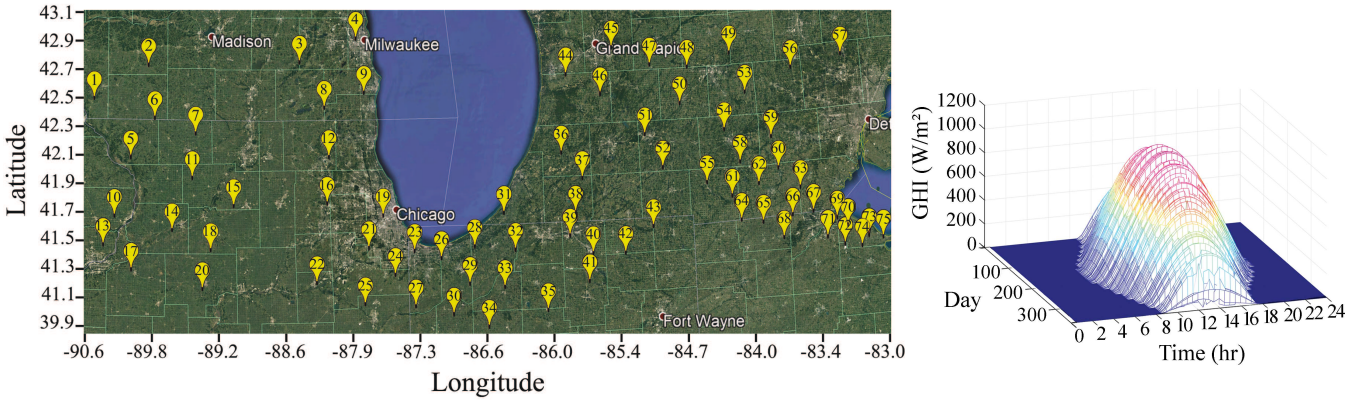
The solar irradiance time series measured at 75 solar sites in northern states of the US near the Lake Michigan are collected in the National Solar Radiation Database (NSRDB) [44] by the National Renewable Energy Laboratory. Fig. 1(a) depicts the latitude-longitude map of solar sites where the spatio-temporal solar radiation data is collected. The data at each site contains the GHI time series with 30-min intervals from 1998 up to 2016. Fig. 1(b) is the plot of GHI values at the solar site 14 in 2015. As shown here, GHI increases from 8:00 to 13:00, and then, decreases until it reaches zero from about 18:00 to 20:00. Generally speaking, we have larger GHI around the day 200 (mid-July), and as we go further, the GHI declines.

The spatio-temporal data is modeled as an undirected graph where each node represents a solar site and each edge reflects the correlation between the corresponding nodes/sites. Let us define a weighted graph $G = (V_G, E_G)$ where V_G is the set of nodes v_i $i = 1, 2, ..., n$ and E_G is the set of edges $e_{k\,l}$ connecting v_k to v_l. The weighted adjacency matrix A is defined by the following formulation:

$$A(k,l) = \begin{cases} \mathbf{e}^{-D(k,l)} & MI\,(k,l) \geq \alpha \\ 0 & MI\,(k,l) < \alpha \end{cases} \quad (1)$$

where $\mathbf{e}$ is the Euler's number, and the edge weight between the nodes v_k and v_l is denoted by $A(k,l)$, while their distance is $D(k,l)$. Also, the normalized mutual information (MI) between the historical GHI measurements of these two nodes is denoted by $MI\,(k,l)$. The edge sparsity parameter $\alpha = 0.8$ acts as a threshold on MI values; that is, for each pair of nodes v_k and v_l, if the corresponding MI exceeds α, we consider an edge $e_{k\,l}$ associated with a weight $e^{-D(k,l)}$ while for the nodes with MI less than α, no edges are considered.

Fig. 2 depicts the MI values corresponding to all pairs of solar sites (i.e. nodes in V_G). Considering the latitude-longitude map in Fig 1(a) and the MI matrix in Fig. 2,



(a) Latitude-Longitude map of 75 solar sites in the NSRDB

(b) Solar Irradiance of 2015 at solar site 14

Fig. 1: Visualization of the solar site locations and GHI measurements in the National Solar Radiation Database

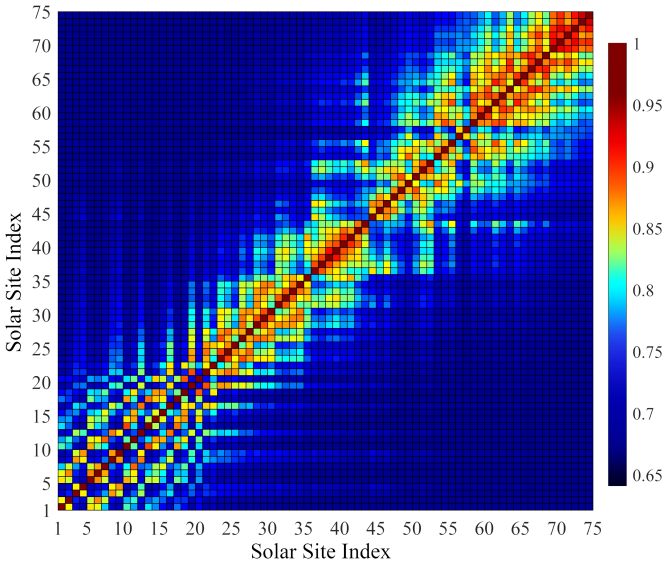


Fig. 2: Mutual Information matrix for all pairs of nodes in V_G. The indices correspond to the indices of solar sites in Fig. 1(a)

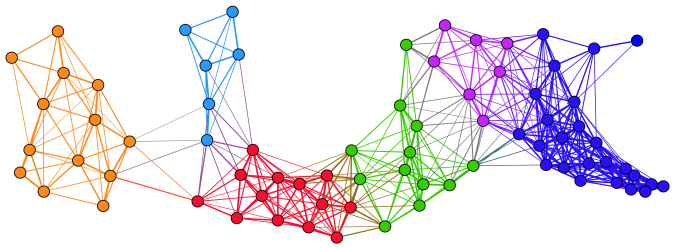


Fig. 3: Structure of the modeled graph G with 75 nodes and 464 edges. The graph is clustered into six Girvan–Newman communities. The width of each edge reflects the magnitude of MI between the corresponding nodes.

we can see that the MI of historical GHI for each pair of sites has high negative correlation with their distance inside the latitude-longitude space. That is, shorter distances lead to higher solar irradiance correlations, which further lead to larger edge weights in the modeled graph G.

Fig. 3 depicts the structure of our graph with 75 nodes and 464 weighted edges clustered into six communities using the Girvan–Newman algorithm [49]. Each community consists of a subset of nodes densely connected to each other with relatively large edge weights due to their high mutual information. The dense edges inside communities and the sparse edges between the communities reflect the strong relationship between the distance of the nodes and their MI.

At each time step t, each node v_i contains a GHI time series $T(v_i, t)$ corresponding to the historical GHI data used as the input to the forecasting model in order to predict some future GHI value ${v_i}^*(t' = t + k)$ with forecast horizon length $k > 0$. The problem is to learn a conditional probability distribution $P^*(V^*(t')|\pi)$ with future GHI tensor $V^*(t') =< {v_1}^*(t'), {v_2}^*(t'), ..., {v_n}^*(t') >$ and historical GHI tensor $\pi =< T(v_1, t), T(v_2, t), ..., T(v_n, t) >$. Considering a training set TS that contains $|TS|$ historical examples $(\pi_j, V_j^*(t'))$ $1 \leq j \leq |TS|$, we need to estimate P^* using the observed π_j and $V_j^*(t')$ in the j-th training example.

The data of 1998-2015 is considered for training our model while the 2016 dataset is used as a test set to evaluate our method. Fig. 4. shows the mutual information between a GHI value at the time $\tilde{t}$ with previous time steps $\tilde{t} - l$ with lag $1 \leq l \leq 300$ for the GHI time series of 1998-2015. As shown in this plot, the GHI values are more correlated with their most recent lags as well as the time lags near $l \in \{24, 48, 72, 96, 120, 144\}$. In this study, in order to make the information in $T(v_i, t)$ useful for the estimation of P^*, we define $T(v_i, t)$ for each node i to be the GHI values corresponding to the lags where the mutual information is equal or greater than some threshold $\tau \geq 0$. Here, τ is a hyperparameter for our model.

III. Proposed Generative Learning Formulation for Nodal Probability Density Estimation in Graphs

A. Generative Learning for PDF approximation

Here, our problem is to capture a probability distribution $P(X)$ over n-dimensional data points X in a potentially high

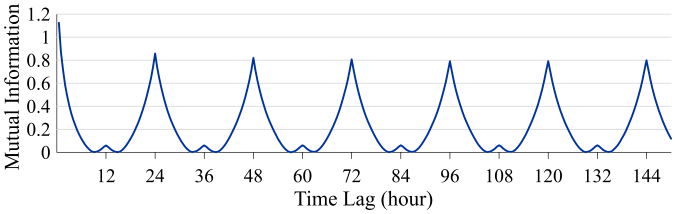


Fig. 4: Mutual Information of future GHI with previous time lags

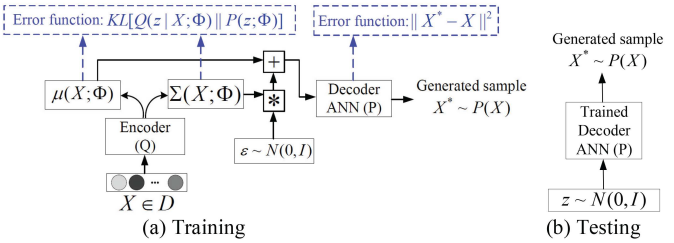


Fig. 5: Structure of CGAE. (a) shows the training process where the model generates $X^* \simeq X$. (b) shows the testing process where the trained decoder generates as many samples $X^* \sim P(X)$ as required simply by feeding a random $z \sim N(0, I)$ to the decoder ANN. The decoder captures PDF $P(X)$.

dimensional vector space $\mathcal{X} \subseteq \mathbb{R}^n$. In fact, we want to be able to generate many samples X^* as close as possible to X. As the complexity of the dependencies between variables of $\mathcal{X}$ grows, the difficulty of learning the true $P(X)$ increases. Hence, we define a "latent variable"-based model in which the hidden random vector $z \in Z$ embodies the major characteristics of $P(X)$ (e.g. the PDF of the future GHI, or any desired nodal PDF in a graph-structured data). More specifically, z is sampled following some unknown distribution $P(z)$ over the high dimensional space Z. To justify that our approach is generative (i.e. the model can generate samples X^*), we ensure that there exists at least one configuration $\hat{z} \in Z$ that causes the model to generate some sample $\hat{X}$ in $\mathcal{X}$. Assuming a family of deterministic functions $f(z;\theta)$ with parameters $\theta \in \Theta$, each "latent variable-parameter" pair is mapped to a sample in $\mathcal{X}$ using $f : Z \times \Theta \to \mathcal{X}$. We find an optimal $\theta^* \in \Theta$ such that when $z \sim P(z)$, the value of $X^* = f(z;\theta = \theta^*)$ is as close as possible to some $X \in \mathcal{X}$. In other words, the probability of f creating an output X^* similar to the observed data X is maximized; hence, our optimization is written as:

$$\theta^* = \underset{\theta}{\arg\max} \left[P(X) = \int f(z;\theta) P(z) dz \right] \tag{2}$$

$f(z)$ is a deterministic function of a random variable z ; hence, for a fixed θ, $f(z;\theta)$ is a random variable in the space $\mathcal{X}$. Therefore, $P(X)$ in (2) can be written as:

$$P(X) = \int P(X|z;\theta) P(z) dz \tag{3}$$

As shown in (2), generating X depends on the latent vector z. Using the Maximum Likelihood framework, if the model converges to the solution θ^*, our generative model is likely to produce X^*. Here, $f(z;\theta)$ is defined as a Gaussian distribution $P(X|z;\theta) = N(X|f(z;\theta), \sigma^2 * I)$ with mean f and a diagonal covariance matrix with entries computed using the hyperparameter σ as the standard deviation. In order to solve the optimization (2)-(3), z should be mathematically defined. Moreover, an estimation for the integral in (2) should be provided. Our main goal is to learn variable z automatically; that is, we opt to avoid describing the dependencies between the dimensions of Z, as no prior knowledge is available/required to solve the problem. Thus, the latent vector is set to $z \sim N(0, I)$ considering Theorem (1):

Theorem (1)**:** *In any space Λ, any complicated probability density function over samples can be modeled using a set of $dim(\Lambda)$ random variables with normal distribution, mapped through a high capacity function.*

As a consequence, an approximator can be learned to map z to some required (desired) hidden variable ξ further mapped to $X \in \mathcal{X}$, to maximize the likelihood of samples X in the dataset D. Here, our f is modeled by an ANN as a standard function approximator capable of learning highly nonlinear target functions using multiple hidden layers. The first layers of these architectures provides a non-linear mapping from $z \in Z$ (with a predefined simple distribution as discussed in this section) to ξ (with an unknown complicated distribution). ξ is further mapped to a sample $X \in \mathcal{X}$ available in D. Notice that if the model has sufficient capacity (ample number of hidden layers, as in the case of deep neural networks), the neural network is able to solve the maximization in (1) to obtain θ^*. Let us rewrite our optimization in (2) using $z \sim N(0, I)$ from Theorem (1):

$$\theta^* = \underset{\theta}{\arg\max} \int N(X|f(z;\theta), \sigma^2 * I) N(z|0, I) dz \tag{4}$$

To solve (4), a distribution function $Q(z|X)$ is defined to decide the importance of an arbitrary configuration $\hat{z} \in Z$ in the generation of a sample X. As a consequence, the expected value of $P(X|z)$ with respect to z, $\mathrm{E}_{z\sim Q}[P(X|z)]$, can be computed using the Kullback–Leibler (KL) divergence:

$$KL[Q(z)||p(z|X)] = \mathrm{E}_{z\sim Q}[\log Q(z) - \log P(z|X)] \tag{5}$$

applying the Bayesian rule for $P(z|X)$, (5) can be written as:

$$\begin{aligned} KL[Q(z)||p(z|X)] &= \mathrm{E}_{z\sim Q}\left[\log Q(z) - \log(\tfrac{P(X|z)P(z)}{P(X)})\right] \\ &= \mathrm{E}_{z\sim Q}[\log Q(z) - \log P(X|z) - \log P(z) + \log P(X)] \end{aligned} \tag{6}$$

This equality is further written as:

$$\begin{aligned} &\log P(X) - KL[Q(z|X)||P(z|X)] \\ &= \mathrm{E}_{z\sim Q}[\log P(X|z) - KL[Q(z|X)||P(z)]] \end{aligned} \tag{7}$$

In order to generate X (that is, create samples $X^* \approx X$), our objective is to maximize $\log P(X)$ while minimizing the KL divergence in the left-hand side of (7); hence, we minimize $\mathrm{E}_{z\sim Q}[\log P(X|z) - KL[Q(z|X)||P(z)]]$ using SGD. Notice that, in the formulation of (7), Q can be viewed as an ANN

encoding X into z, while P is an ANN decoding z to obtain X. To solve the optimization, Q is defined as:

$$Q(z|X) = N(z|\mu(X;\Phi), \Sigma(X;\Phi)) \tag{8}$$

with deterministic functions μ and Σ defined by an ANN with free parameters set Φ trained by SGD. As Q and P are both dimensional multivariate Gaussian distributions, the term $KL[Q(z|X)||P(z)]$ in (7) is computed by:

$$\begin{aligned} &KL\left[Q(z|X)||P(z)\right] \\ &= KL\left[N(z|\mu(X;\Phi), \Sigma(X;\Phi))||N(0,I)\right] \\ &= \tfrac{1}{2}\left[\log\tfrac{\det(I)}{\det(\Sigma)} - d + tr(\Sigma) + (0-\mu)^T(0-\mu)\right] \\ &= \tfrac{1}{2}\left[-\log(\det(\Sigma)) - d + tr(\Sigma) + \mu^T\mu\right] \end{aligned} \tag{9}$$

Therefore, in order to optimize (7), the following optimization problem is solved:

$$\theta^* = {}^{\arg\max}_{\theta} \mathrm{E}_{X\sim D}\left[\begin{array}{l} \mathrm{E}_{z\sim Q}[\log P(X|z;\Phi)] \\ -KL[Q(z|X;\Phi)||P(z;\Phi)] \end{array}\right] \tag{10}$$

Applying the reparametrization technique, (10) can be written as:

$$\theta^* = {}^{\arg\max}_{\theta} E_{X\sim D}\left[\begin{array}{l} E_{\varepsilon\sim N(0,I)}\left[\begin{array}{l} \log P\big(X|z=\mu(X) \\ +\Sigma^{1/2}(X)*\varepsilon\,;\Phi\big) \end{array}\right] \\ -KL[Q(z|X;\Phi)||P(z;\Phi)] \end{array}\right] \tag{11}$$

Fig. 5(a) shows the training structure of our generative model based on (8) and (11) to generate $X^* \approx X$. The encoder ANN, Q, takes X observed in dataset D and outputs μ and Σ (see (8)). The error of the encoder ANN is $KL[Q(z|X)||P(z)]$ computed in (9). The gradient of this error function is used by Stochastic Gradient Descent (SGD) method to train this ANN. After computing μ and Σ using Q, our latent variable $z = \mu(X;\Phi) + \Sigma^{1/2}(X;\Phi) * \varepsilon$ is obtained using (11). Then, z is fed to the decoder ANN, P, to obtain our generated sample $X^* \approx X$. The error function of this ANN is computed by $||X - X^*||^2$ to reflect the distance between the generated sample X^* and its true (observed) value X. When Q and P are trained by SGD, in order to generate a new sample $X^* \approx X$, one can simply feed some $z \sim N(0, I)$ to P and obtain X^* as shown in Fig. 5(b).

B. Convolutional Graph Autoencoder

In Section III-A, our objective was to learn $P(X)$ in some high dimensional space $\mathcal{X}$ by generating $X^* \approx X$. Here, we aim to learn $P^*(V^*|\pi)$, i.e. PDF of V^* in G given π. We present our CGAE shown in Fig. 6 as the first generative model that captures nodal distribution $P^*(V^*(t')|\pi)$ in a graph G. Given historical GHI π, our objective is to generate ρ samples $\hat{V} \approx V^*$ to estimate $P^*(V^*|\pi)$.

Let us mathematically formalize how CGAE generates $\hat{V}$ as an estimation for V^* :

$$\hat{V} = \mu(\pi, z) + \varepsilon \text{ s.t. } z \sim N(0,1)\,,\ \varepsilon \sim N(0,1) \tag{12}$$

both z and ε are white Gaussian noises. μ is implemented by an ANN as in Section III-A. Assuming $z \sim Q$ using PDF $Q(z)$, Bayes rule [50] is applied to compute $\mathrm{E}_{z\sim Q}[\log P(V^*(t')|z,\pi)]$:

$$\begin{aligned} \mathrm{E}_{z\sim Q}[\log P(V^*(t')|z,\pi)] = \mathrm{E}_{z\sim Q}[\log P(z|V^*(t'),\pi) - \\ \log P(z|\pi) + \log P(V^*(t')|\pi)] \end{aligned} \tag{13}$$

(13) is rewritten as:

$$\begin{aligned} \log P(V^*(t')|\pi) - \mathrm{E}_{z\sim Q}[\log Q(z) - \log P(z|\pi, V^*(t')) = \\ \mathrm{E}_{z\sim Q}[\log P(V^*(t')|z,\pi) + \log P(z|\pi) - \log Q(z)] \end{aligned} \tag{14}$$

Now, following (8), we have $Q = N(\mu'(\pi, V^*(t'))\ ,\ \sigma'(\pi, V^*(t')))$ where μ' and σ' are ANNs trained alongside μ. Let us denote Q by $Q(z|\pi, V^*)$, (14) is written as:

$$\begin{aligned} \log P(V^*|\pi) - KL[Q(z|\pi,V^*)||P(z|\pi,V^*)] = \\ \mathrm{E}_{z\sim Q}[\log P(V^*|z,\pi)] - KL[Q(z|\pi,V^*)||P(z|\pi)] \end{aligned} \tag{15}$$

Considering (15), our objective is to increase $E_1 = \log P(V^*|z,\pi)$ and $E_2 = -KL[Q(z|\pi,V^*)||P(z|\pi)]$. CGAE is trained by SGD to maximize $E_T = E_1 + E_2$. This leads to maximizing the likelihood of V^* while training Q to accurately estimate $P(z|\pi, V^*)$. Note that, similar to our optimization in Section III-A, we have $P(z|\pi) = N(0,1)$. Our latent vector is $z = \mu'(\pi, V^*(t')) + \alpha \circ \sigma'(\pi, V^*(t'))$ where $\alpha \sim \mathrm{N}(0,1)$ and $\circ$ is the element-wise product operation. E_T is differentiable with respect to the whole parameters of CGAE (including the parameters in ANNs corresponding to μ, μ' and σ'); hence, the whole CGAE model can be easily tuned by SGD to maximize E_T. In Section III-C, the neural architecture corresponding to our CGAE is defined based on ANNs.

C. CGAE Architecture

CGAE consists of three ANNs; 1- Graph Feature Extraction ANN, which gives us a compact representation of π stored in G, denoted by $R(G)$, 2- Encoder ANN, Q, that implements μ' and σ' to capture $Q(z|\pi, V^*)$, and 3- Decoder ANN, P, that implements $\mu(\pi, z)$ in (12), to produce samples $\hat{V}$ drawn from the true future GHI distribution $P^*(V^*(t')|\pi)$.

a) Graph Feature Extraction ANN (Computing R(G)): At each training step t, the spectral graph convolutions of G, which stores $\pi =< T(v_1,t), T(v_2,t), ..., T(v_n,t) >$ inside its nodes, is computed by $\psi_\theta * \pi = U\psi_\theta U^T\pi$. Here, U is the eigenvector matrix of the normalized Laplacian $L = U\Omega U^T$ and $\theta \in \mathrm{R}^n$ is the parameter vector for the convolutional filter $\psi_\theta = diag(\theta)$ in the Fourier domain. Notice that the Fourier transformation of π is computed by $U^T\pi$. ψ_θ is defined as a function of L 's eigenvalues; hence, our filter is denoted by $\psi_\theta(\Omega)$. Estimating $\psi_\theta(\Omega)$ by Chebyshev Polynomials [51], [52] P_j, we have $\psi_\omega \approx \sum_{j=0}^{J} \omega_j P_j(\frac{2}{\gamma_{\max}}\Omega - I)$ where $\gamma_{\max}$ is the maximum eigenvalue of L, and ω_j is the j -th Chebyshev coefficient. Therefore, the spectral graph convolution function on G is:

$$\psi_\omega * \pi \approx \sum_{j=0}^{J} \omega_j P_j(\frac{2}{\gamma_{\max}}\,\Omega - I)\pi \tag{16}$$

The convolution in (16) is further simplified by $\delta = \omega_0 = -\omega_1$ which decreases parameters' size while $\gamma_{\max} = 2$ for

$J = 1$; As a result, (16) can be computed by:

$$\psi_\omega * \pi \approx \omega_0 P_0(L-I)\pi + \omega_1 P_1(L-I)\pi = \delta(I + D^{-\frac{1}{2}} A D^{-\frac{1}{2}})\pi \quad (17)$$

Based on the convolution (17), a graph feature extraction neural network (GFENN) with L_G hidden layers is defined to extract spatio-temporal features from GHI observations at all nodes/sites of G. Here, the output of each layer $1 \leq k \leq L_G$ is:

$$O^k = \mathrm{ReLU}(MO^{k-1}\mathrm{W}^k) s.t. M = \tilde{D}^{-\frac{1}{2}}(A+I)\tilde{D}^{-\frac{1}{2}} \quad (18)$$

where $\tilde{D}_{ii} = \sum_j (A+I)_{ij}$. The input of GFENN is $O^0 = \pi$ while the output is G 's spatio-temporal representation $R(G) = O^{L_G}$.

b) The encoder (Q) and Decoder (P): Since GFENN captures spatiotemporal features of π, and stores them in $R(G)$, one can view CGAE as a model estimating $P^*(V^*|R(G))$ instead of $P^*(V^*|\pi)$. In Section III-A, (8) showed that Q can be viewed as an ANN encoding input tensor X into the latent vector z while P is a decoding ANN that maps z to X. As depicted in Fig. 6, Here, the input to the encoder Q is $X = R(G)$. Our encoder Q is defined by a deep ANN with L_Q hidden layers and ReLU activations for each hidden layer, trained to encode V^* into a latent vector $z \in Z$, such that the resulting z can be decoded back to V^*. As discussed in (15) and also shown in Fig. 6, the error function for the encoder Q is defined by:

$$\begin{aligned} Err_Q &= KL[Q(z|\pi, V^*)||N(0,1)] \\ &= KL[Q(z|R(G), V^*)||N(0,1)] \end{aligned} \quad (19)$$

Similar to Q, our decoder, P, is implemented by a deep ANN with L_P hidden layers using ReLU activations to take the latent vector z learned by Q, as well as the graph representation $R(G)$, and decode them to generate an approximation of V^*, denoted by $\hat{V}$. To make the generated sample $\hat{V}(t')$, as close as possible to the real future value $V^*(t')$ we minimize the following reconstruction error for P :

$$Err_P = ||V^*(t') - \hat{V}(t')||^2 \quad (20)$$

Therefore, the total error optimized by the stochastic gradient descent method is $E = Err_Q + Err_P$.

D. Estimation of $P(V^*|\pi)$

As shown in Fig. 6(b), during test time, $R(G)$ and $z \sim N(0, I)$ are fed to the decoder ANN and the estimation $\hat{V}(t')$ is obtained. No encoding is needed; hence, generating estimations $\hat{V}(t') \approx V^*(t')$ is dramatically fast. All we need to do to generate a new sample $\hat{V}(t')$, is to sample a new $z \sim N(0, I)$ and run feed-forward algorithm on the GFENN (to obtain $R(G)$) and the decoder ANN (to obtain the desired result, i.e. $\hat{V}(t')$). Following this approach, we generate ρ number of samples $\hat{V} \sim P(V^*|\pi)$ to estimates $P(V^*|\pi)$ using the decoder. As a result, our decoder P generates the PDF of future GHI mapping $N(0, I)$ to $P(V^*|\pi)$.

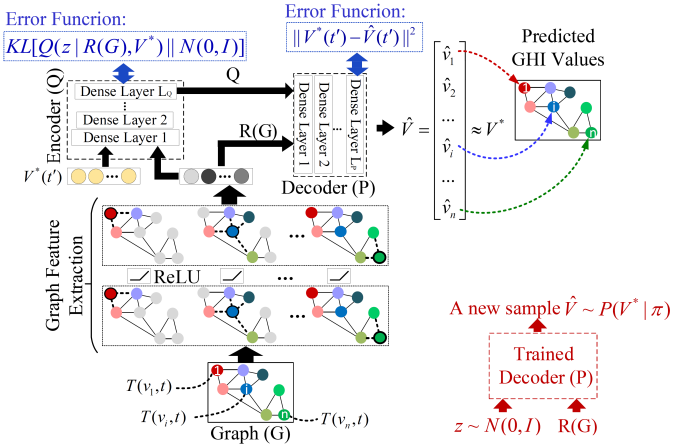


Fig. 6: Convolutional Graph Autoencoder

IV. Numerical Results

CGAE is compared with recent temporal as well as spatio-temporal benchmarks utilized for short-term irradiance/PV probabilistic forecasting. The temporal models include Quantile Regression (QR) [21], Kernel Density Estimation (KDE) [20], Extreme Learning Machines (ELM) [18], and Probabilistic Persistence (PP) [53], while the spatio-temporal benchmarks include the Space-time Copula [45], Spatio-Temporal QR-Lasso [46], Compressive Spatio-Temporal Forecasting [47], and Spatio-Temporal Support Vector Regression [17], [48]. The advantages of spatio-temporal feature learning for the underlying problem is shown. Since no generative model was presented in the literature, the experiments motivate further research on generative modeling for renewable resources prediction.

A. Experimental Settings

As explained in Section II, the NSRD dataset is applied to train/test our model. The 1998-2015 data is used to train CGAE while the 2016 data is applied to evaluate the prediction performance. In this study, CGAE is trained/tested to forecast GHI time series from 30 min (horizon length $k = 1$) up to 6 hours ahead ($k = 12$). Batch Gradient Descent with learning rate $\eta = 5 * 10^{-4}$ is employed to train our CGAE (including GFENN, encoder ANN, and decoder ANN) by minimizing the error $Err_Q + Err_P$ using batch size k equal to 400. In this study, the number of generated samples is $\rho = 10^4$, and the number of GFENN layers is set to $L_G = 2$ while $L_P = 4$ and $L_Q = 3$. The feature selection hyperparameter is $\tau = 0.45$.

We employed the Information Theoretical Estimators (ITE) library [54] to compute the mutual information matrix corresponding to the historical GHI time series in Section II. The ITE is used as a free and open source toolbox in Matlab 2018. The graph modeling process of Section II is implemented in Gephi 0.9.2 [55] which is an open-source software for network visualization and analysis. Moreover, our proposed deep neural network, CGAE, is implemented in Python 3.6 with Keras 2.2.4 library [56] and GPU-based Tensorflow 1.7.0 [57] backend. The model is implemented on a computer system with Intel Core-i7 4.1GHz CPU and NVIDIA GeForce

GTX 1080-Ti GPU. Our GPU supports CUDA 9.0 which is a parallel computing platform that helps Tensorflow to speed up all the computations in Keras.

B. *Performance Comparison (Quantitative Results)*

The prediction quantiles of our model are compared with both temporal and spatio-temporal methodologies in terms of reliability, sharpness and Continuous Ranked Probability Score (CRPS):

1) Reliability: This criterion shows how closely the prediction probabilities correspond to the observed (real) frequencies of the GHI data. Here, the bias $R^{1-2\alpha}$ is computed by:

$$R^{1-2\alpha} = \left(\frac{N^{1-2\alpha}}{N} - (1-2\alpha) \right) \times 100\% \qquad (21)$$

where N is the number of test examples, $N^{1-2\alpha}$ is the number of observations covered by the nominal coverage rate $(1-2\alpha)\times 100\%$. The closer the nominal coverage of prediction intervals is to the observed (actual) coverage rate, the higher the reliability is; hence, small $R^{1-2\alpha}$ shows better accuracy. In fact, $R^{1-2\alpha} = 0$ corresponds to the perfect (ideal) reliability.

Fig. 7 depicts the reliability measurements averaged over all GHI nodes/sites with various nominal coverage rates ranging from 10% to 90%. As shown in this figure, the spatio-temporal prediction models including CGAE, ST-Copula, ST-QR-Lasso, CSTF, and ST-SVR, lead to more reliable probabilistic forecasts compared to the temporal models such as ELM, KDE, QR, and PP. For instance, the ST-QR-Lasso model which is a spatio-temporal version of QR, leads to an average deviation of 5.46% while the QR obtains 9.13% deviation compared to the ideal prediction model with zero deviation. Among the temporal models, PP has the worst reliability which results in the largest average deviation equal to 10.62%. ELM leads to the highest reliability among temporal models with 6.71% absolute deviation. This model yields 36.81%, 26.49%, and 22.59% more reliable (less deviated) predictions compared to PP, QR, and KDE, respectively. The major reason for this observation is the better generalization of neural network-based approaches compared to the traditional statistical approaches. In contrast to other temporal benchmarks, ELM has a large nonlinear parameter space which helps this model to improve generalization and obtain more reliable outcomes. Our deep learning-based generative model, CGAE, outperforms all temporal benchmarks, with 86.35%, 84.12%, 83.28%, and 78.40% better reliability compared to PP, QR, KDE, and ELM. The smaller deviation of CGAE compared to ELM is mainly due to CGAE's graph-based spatial feature extraction as well as its larger hypothesis space caused by the higher number of nonlinear computational layers.

Among the spatio-temporal prediction benchmarks, CGAE and ST-SVR have the least deviated predictions with 1.45% and 4.02% average absolute deviations, respectively. The reliable performance of ST-SVR is due to its ability to handle complex high-dimensional feature spaces using the kernel trick. The smaller deviation of CGAE in comparison with other spatio-temporal benchmarks shows the effectiveness of

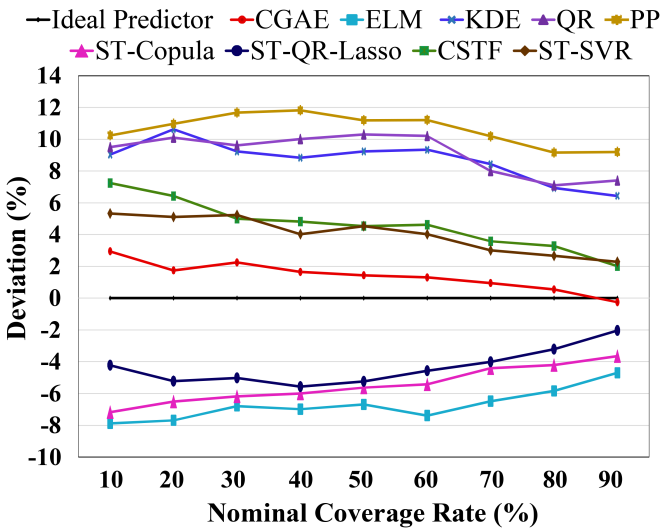


Fig. 7: Reliability measurements averaged over all GHI nodes/sites

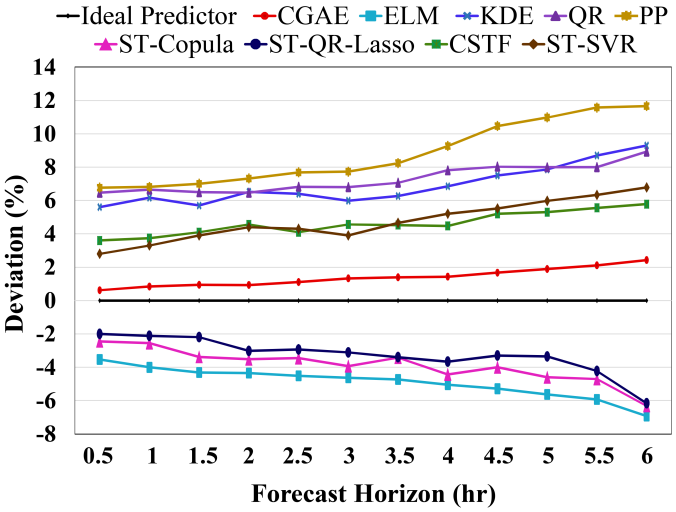


Fig. 8: Average reliability with different look-ahead times

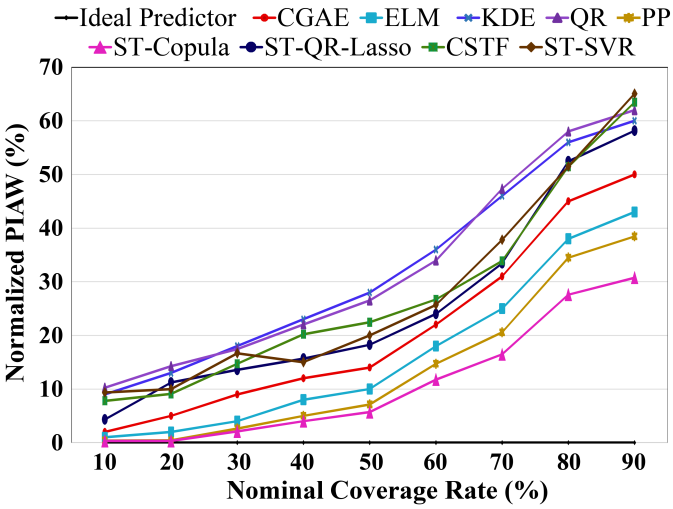
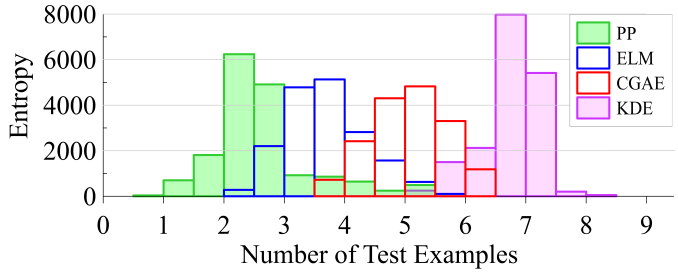
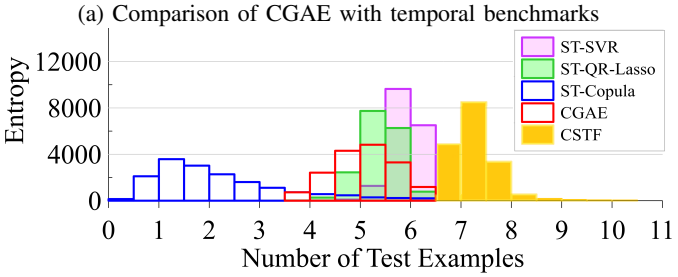


Fig. 9: Sharpness evaluation using normalized PIAW



(a) Comparison of CGAE with temporal benchmarks



(b) Comparison of CGAE with spatio-temporal benchmarks

Fig. 10: Entropy diagram of CGAE with various temporal and spatio-temporal benchmarks for the 6-hr ahead forecasts

our GFENN in providing powerful spatial information from the underlying solar sites.

Fig. 8 shows the average reliability with different look-ahead times for various temporal and spatio-temporal benchmarks. As shown in this plot, the slope of the deviation curve for all benchmarks start to increase significantly from the 3.5-hr horizon, while CGAE has a much smaller slope. As the time horizon expands, the improvement of CGAE becomes more significant. PP has the worst performance, especially in longer horizons, compared to other methodologies. This is due to its low generalization capacity resulted from its smoothness assumption of the target function, which undermines its efficiency in practice. The spatio-temporal approaches have less than 6.31% deviation for all time horizons while even the most reliable temporal model, ELM, exceeds this limit for 5.5-hour and 6-hour ahead predictions. CGAE yields 1.10% and 4.49% better reliability in 3-hr and 6-hr forecasts compared to ELM, respectively. This shows the superiority of generative modeling over discriminative modeling introduced in previous ANN methods in the literature. The relatively small deviation of spatio-temporal models is resulted by their good unbiased prediction, while temporal models are more biased, which degrades their efficiency in practical applications.

Among the spatio-temporal approaches, the CGAE, CSTF, and ST-SVR have smaller deviation slope with respect to the time horizon. While ST-QR-Lasso and ST-Copula have a significant growth in their deviation slope after the 5-hr time horizon, the CGAE, CSTF, and ST-SVR show a smooth deviation curve with a relatively small gradient. As shown in Fig. 8, CGAE shows more reliable predictions in comparison with all spatio-temporal benchmarks. As the time horizon expands, the superiority of CGAE becomes more noticeable. For the 6-hr ahead prediction, CGAE obtains 4.35%, 3.88%, 3.72%, and 3.35% better reliability in terms of the deviation from the ideal prediction compared to ST-SVR, ST-Copula, ST-QR-Lasso, and CSTF, respectively.

2) Sharpness:: Sharpness is a complementary metric to the reliability, which evaluates the concentration of the prediction distribution. The criterion shows how informative a forecast is by narrowing down the predicted GHI values. Sharpness should be analyzed with respect to reliability, as high sharpness does not necessarily show better prediction when the model has low reliability (high deviation in Fig. 7 and Fig. 8). Sharpness is investigated using two performance metrics:

a) Prediction Interval Average Width (PIAW): This metric, $PIAW^{\alpha}$, evaluates sharpness for the nominal coverage rate $(1-2\alpha)\times 100\%$ by:

$$PIAW_{\alpha}=\frac{1}{N}\sum_{n=1}^{N}|q^{\alpha}(n)-q^{1-\alpha}(n)| \tag{22}$$

where $q^{\alpha}(n)$ and $q^{1-\alpha}(n)$ represent the α and $1-\alpha$ prediction quantiles for the n -th test sample. Fig. 9 shows the average sharpness of 10%-90% nominal coverage rates normalized by maximum observed GHI. As shown in this diagram, among temporal models, PP has the sharpest intervals in all nominal coverage rates; however, as shown by Fig. 7 and Fig. 8, it has poor reliability compared to other benchmarks especially when the horizon is expanded. Moreover, ELM provides overly narrow quantiles leading to higher sharpness compared to CGAE. However, such high sharpness does not contribute to forecast accuracy/reliability. Large amount of sharpness might work in the case of clear sky when no significant uncertainty is present and GHI is predictable with high accuracy; however, in other cases (e.g. when GHI is varying during a rainy day), it would lead to poor performance as the model would neglect the risk of uncertainties in GHI. CGAE provides medium sharpness which is not too high to lead to erroneously narrow quantiles (as in the case of PP and ELM), and not too low to lose information about future GHI (as in the case of KDE and QR).

Generally speaking, the spatio-temproal models obtain moderate sharpness values that are neither as high as KDE nor as low as PP. Among this category of models, ST-Copula is an exception which provides prediction intervals even sharper than the PP. The sharpness metric shows that ST-Copula is likely to provide biased predictions that are over-confident. In practice, such confidence can lead to poor performance since the reliability of ST-Copula is lower than the other spatio-temporal benchmarks. As shown by Fig. 9, the ST-QR-Lasso, CSTF, and ST-SVR provide similar sharpness for 60% and 70% nominal coverage rates; however, for other coverage values, the prediction intervals of ST-QR-Lasso, CSTF, and ST-SVR become too sharp while CGAE maintains its moderate sharpness.

b) PDF Entropy: The sharpness of a forecast can be estimated using the entropy of the prediction PDF. Sharper forecasts lead to smaller PDF entropies. Fig. 10(a) shows the histogram of the entropies of all temporal benchmarks for the 6-hr ahead prediction task. As shown in this plot, the majority of forecasting PDFs for PP and ELM correspond to low values. The mean entropy of PP and ELM are 2.77

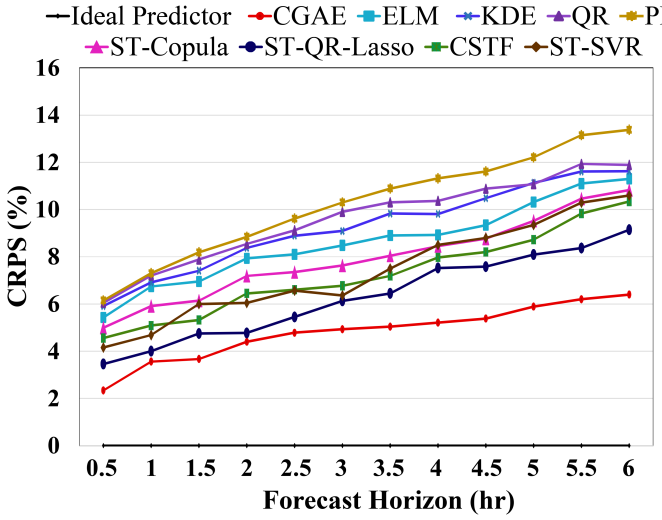


Fig. 11: CRPS results of 30-min up to 6-hr ahead predictions

and 3.69, respectively. The low entropy of PP is due to the consecutive clear days in the testing set where the variance of the prediction PDF is small. Such small entropies/variances result in overconfident predictions caused by the lack of knowledge about future GHI uncertainties. The overly narrow prediction quantiles in ELM lead to low PDF entropies which degrade accuracy since the uncertainties in the future GHI are disregarded by predictions less reliable than CGAE (see Fig. 7 and Fig.8). CGAE has moderate sharpness and medium entropy values with mean 5.15. KDE has high entropies with mean 6.77 and a small variance of 0.22 that result in high uncertainty boundaries for the future GHI and less informative forecasts compared to CGAE and ELM. In contrast to ELM and KDE, our CGAE model has entropies that are not too low (as in the case of ELM) to disregard GHI uncertainties and not too high (as in the case of KDE) to provide under-confident predictions.

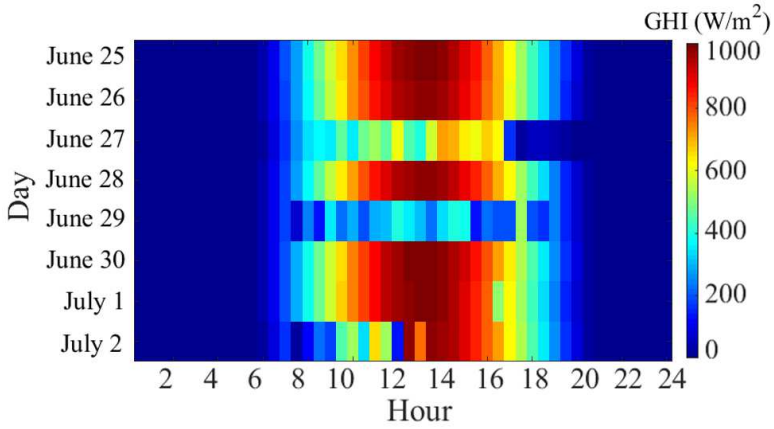
Fig. 10(b) depicts the histogram of the entropies of all temporal benchmarks for the 6-hr ahead prediction task. As shown in this diagram, ST-Capula obtains relatively small entropy which is reflected by the over-confidence and large bias in the prediction PDFs of this model. On the other hand, the CSTF leads to under-confident results with high entropies. The mean entropy of CSTF is 7.26 which is 19.01%, 23.83%, and 29.06% higher than the ST-SVR, ST-QR-Lasso, and CGAE, respectively. This is mainly due to having high variance (high uncertainty) in consecutive sunny days when predicting by CSTF. Such variance is degraded by ST-SVR, ST-QR-Lasso, and CGAE as these models provide a better bias (larger bias) when they encounter multiple consecutive sunny days in the test set. The moderate entropy obtained by CGAE shows that this model is not too biased (as in the case of ST-Copula) to neglect GHI uncertainties in the dataset, and not too uncertain (as in the case of CSTF) to provide uninformative predictions with high unreliability.

3) Continuous Ranked Probability Score: CPRS is a metric evaluating the entire prediction distribution reflecting the deviations between the CDF of the predicted and observed data. One can view CRPS as a metric combining reliability and sharpness to provide a comprehensive performance evaluation. CRPS is computed by:

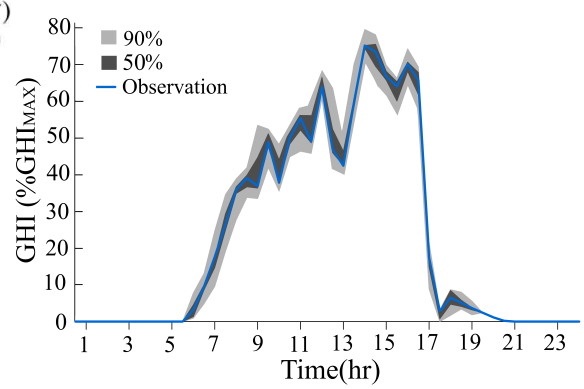
$$CRPS(F,v) = \int_{-\infty}^{\infty} (F(x) - U(x-v))^2 dx$$
$$s.t.\ U(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (23)$$

with the prediction CDF F and the Heaviside function U. The average CRPS of all benchmarks for 30-min up to 6-hr ahead GHI forecast is depicted in Fig. 11. The smaller CRPS a model obtains, the better the accuracy it provides. As shown in this plot, the ANN-based methodologies, ELM and CAGE, outperform the temporal methods PP, QR, and KDE. ELM achieves 1.24% and 1.38% better CRPS on average over all time horizons compared to KDE and QR, respectively. KDE has slightly better performance in comparison with QR for 30-min up to 2.5-hr ahead predictions. The better accuracy of KDE becomes more noticeable in the horizon range of 3 hr up to 4.5 hr. Similar superiority is reflected by the better reliability curve of KDE compared to QR in Fig. 8. Among all temporal benchmarks, PP has the worst performance. This model has 1.77% and 1.49% more CRPS on average for 6-hr prediction, compared to KDE and QR, respectively. As the forecast horizon length grows, the CRPS of PP increases by larger amounts compared to other benchmarks. This is due to low generalization capability and erroneously high sharpness (low entropy as shown in Fig. 10(a)) which results in unreliable predictions, especially when the weather condition changes from sunny to cloudy since this approach suffers from the naïve smoothness assumption. As depicted in Fig. 11, CGAE shows better performance in comparison with all temporal models because of its high reliability (shown by Fig. 7 and Fig. 8) and appropriate sharpness (i.e., moderate PIAW and entropy in Fig. 9 and Fig. 10). CGAE outperforms ELM by 2.98% CRPS for hourly prediction, which is increased significantly for time horizons of length more than 3 hours and reaches the 4.90% CRPS improvement for 6-hr ahead predictions.

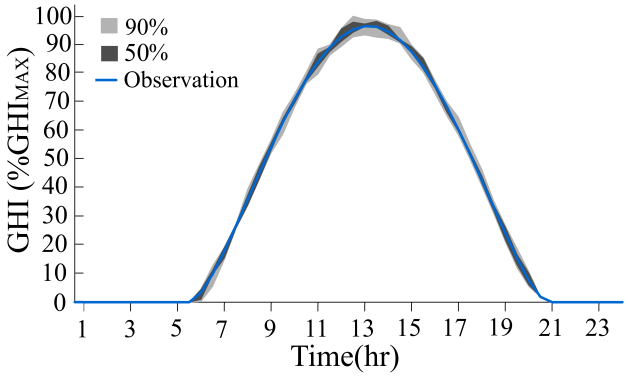
The spatio-temporal models generally have smaller CRPS due to modeling the spatial behavior of GHI observations as well as the temporal characteristics. For instance, the ST-Copula leads to 1.29% CRPS improvement compared to ELM for hourly predictions. Moreover, the ST-QR-Lasso model obtains 3.29% better average CRPS over all time horizons compared to its temporal version i.e. QR. While CSTF and ST-SVR obtain close CRPS curves especially for time horizons longer than 4 hours, the ST-QR-Lasso significantly dominates with lower CRPS values. The better performance of ST-QR-Lasso is mainly due to directly handling the high dimensionality and over-fitting issues that characterize the use of large amounts of data. In fact, the Lasso technique is very useful to reduce the likelihood of overfitting for most practical applications where a large number of observations are available. CGAE obtains 2.53% better CRPS in comparison with the ST-QR-Lasso. Although both models use L1-regularization techniques to avoid overfitting, the CGAE model obtains better



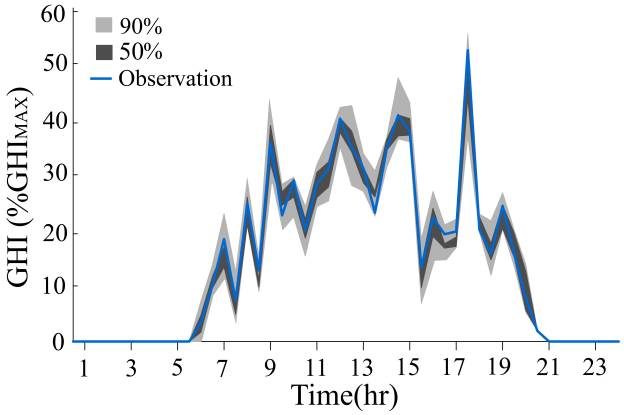
(a) Observed GHI data from June 25^{th} to July 2^{nd}



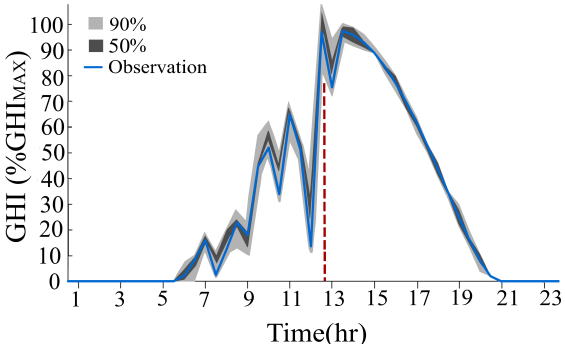
(b) June 27^{th}



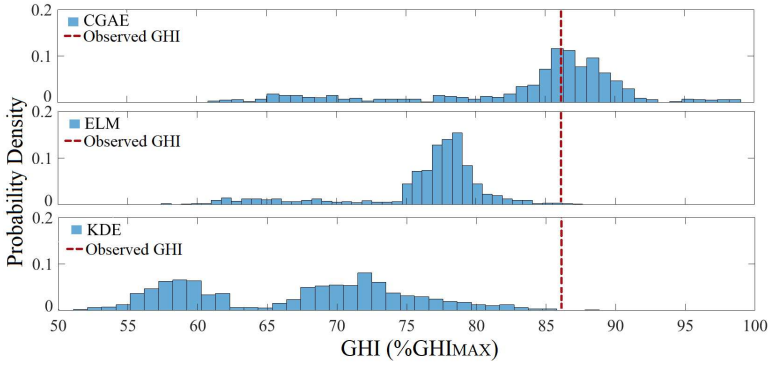
(c) June 28^{th}



(d) June 29^{th}



(e) July 2^{nd}



(f) Histogram of predicted GHI for July 2^{nd} 12:30

Fig. 12: Predicted densities forecasted by CGAE in four days between June 25^{th} and July 2^{nd} with various weather conditions

accuracy due to providing a very large hypothesis space which leads to better generalization capacity.

C. *Qualitative Results*

The probabilistic prediction of CGAE is investigated to show the capability of our model under different weather conditions. Fig. 12(a) shows the GHI values of eight days, from June 25^{th} to July 2^{nd} in 2016, for a site near the Michigan Lake. As shown in this plot, the selected days contain various weather conditions including sunny, partly cloudy, and overcast, in a short period of time. June 25^{th} and 26^{th} are both sunny with high GHI, while the subsequent day, June 27^{th}, is mostly cloudy with many variations. The next day, June 28^{th} is sunny with high GHI while June 29^{th} is overcast with very small irradiance. June 30^{th} and July 1^{st} are sunny, and the last day, July 2^{nd} is a combination of partly cloudy and sunny. This test case evaluates the performance of CGAE when the weather changes dramatically from one day to the other, and within each day. As shown in Fig. 12 (b)-(e), the prediction intervals of CGAE with 50% and 90% confidence rates follow the actual GHI values with high accuracy resulting in good reliability. In Fig. 12(b), as the weather changes from sunny to partly cloudy around 9:00, the confidence boundaries expand showing the increase in the

prediction uncertainty. In Fig. 12(c), June 28^{th} has a very smooth GHI curve measured on a clear sunny day, hence, the model's uncertainty is very small. In Fig. 12(d)-(e) the weather has significant changes during overcast in June 29^{th} and partly cloudy and sunny conditions in July 2^{nd}. As seen in these two figures, although the uncertainty is increased in such conditions, the model still follows the observed GHI with high reliability. On July 2^{nd}, at 12:30, the GHI jumps drastically from 12% of maximum GHI, GHIMAX, to 86%. Fig. 12(f) shows the histogram of the predicted GHI for this observation. As shown in this figure, CGAE could capture this jump more reliably having heavier probability density around 85%-90% GHIMAX. However, ELM and KDE assign a high probability to smaller values as these models are more affected by previous small measurements. Moreover, KDE does not provide enough sharpness for this example, hence, its prediction cannot be informative. Having much higher generalization capability and being able to leverage spatio-temporal information from GHI observations, our CGAE can capture uncertainties in the solar data with higher accuracy and appropriate sharpness.

D. Running Time Analysis

As mentioned in Section IV-A, our proposed model, CGAE, is trained offline using the batch gradient descent method. In the batch gradient descent with batch size k, the gradients of the error function with respect to k training samples are aggregated in each batch at each training iteration; therefore, increasing the batch size k would lead to an increase in the training speed. Fig. 13 depicts the effect of batch size on the training time of CGAE for the prediction tasks with different time horizons. As shown in this figure, the running time decreases with the increase of batch size. For instance, in the 1-hr ahead prediction task, $k = 50$ leads to a training time equal to 21.39 min, while using $k = 400$ takes 19.90 min.

Fig. 13 also shows the effect of the forecast horizon in the training time of the proposed model. As shown in this figure, for a fixed k, the training time increases as the time horizon is extended. For instance, when $k = 200$ CGAE takes 20.33 min to train its parameters for the 1-hr ahead prediction task, while the training time increases to 25.32 min for 6-hr ahead forecasts.

As discussed in Section III-D, CGAE uses a simple feed-forward approach during the testing time; therefore, our model leads to fast predictions. The average testing time of CGAE for all forecast time horizons is less than 0.35 sec; hence, the proposed approach can be effectively used for all real-world applications.

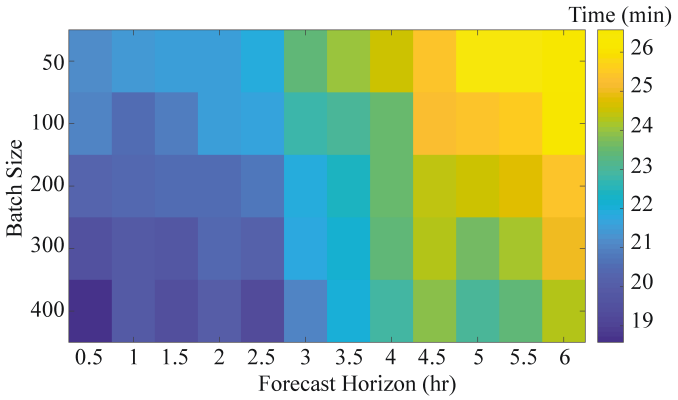


Fig. 13: Running time of the CGAE using various batch size values

V. Conclusions

A novel deep generative model, Convolutional Graph Autoencoder, is presented for a new problem, nodal distribution learning in graphs. The model captures deep convolutional features from an arbitrary graph-structured data, to learn the corresponding probability densities of nodes. Here, the problem of spatio-temporal solar irradiance forecasting is presented as a graph distribution learning problem where each node of the graph represents a solar irradiance measurement site, while each edge represents the distance between the sites. Using graph spectral convolutions, the spatial features of the solar data are extracted, that are further used by an encoding and decoding ANN to capture the distribution of future solar irradiance. Our deep learning model is used to provide probabilistic forecasts for the National Solar Radiation Database. Simulation results show better reliability, sharpness and Continuous Ranked Probability Score compared to recent baselines in the literature.

References

[1] C. Voyant, G. Notton, S. Kalogirou, M.-L. Nivet, C. Paoli, F. Motte, and A. Fouilloy, "Machine learning methods for solar radiation forecasting: A review," *Renewable Energy*, vol. 105, pp. 569–582, 2017.

[2] C. Wan, J. Zhao, Y. Song, Z. Xu, J. Lin, and Z. Hu, "Photovoltaic and solar power forecasting for smart grid energy management," *CSEE Journal of Power and Energy Systems*, vol. 1, no. 4, pp. 38–46, 2015.

[3] M. Khodayar, O. Kaynak, and M. E. Khodayar, "Rough deep neural architecture for short-term wind speed forecasting," *IEEE Trans. Ind. Inform*, vol. 13, pp. 2770–2779, 2017.

[4] Y. Jiang, H. Long, Z. Zhang, and Z. Song, "Day-ahead prediction of bihourly solar radiance with a markov switch approach," *IEEE Transactions on Sustainable Energy*, vol. 8, no. 4, pp. 1536–1547, 2017.

[5] H. C. Hottel, "A simple model for estimating the transmittance of direct solar radiation through clear atmospheres," *Solar energy*, vol. 18, no. 2, pp. 129–134, 1976.

[6] S. Pfenninger and I. Staffell, "Long-term patterns of european pv output using 30 years of validated hourly reanalysis and satellite data," *Energy*, vol. 114, pp. 1251–1265, 2016.

[7] D. P. Larson, L. Nonnenmacher, and C. F. Coimbra, "Day-ahead forecasting of solar power output from photovoltaic plants in the american southwest," *Renewable Energy*, vol. 91, pp. 11–20, 2016.

[8] J. Nou, R. Chauvin, S. Thil, J. Eynard, and S. Grieu, "Clear-sky irradiance model for real-time sky imager application," *Energy Procedia*, vol. 69, pp. 1999–2008, 2015.

[9] W. Liu, C. Liu, Y. Lin, L. Ma, F. Xiong, and J. Li, "Ultra-short-term forecast of photovoltaic output power under fog and haze weather," *Energies*, vol. 11, no. 3, p. 528, 2018.

[10] S. Cros, O. Liandrat, N. Sébastien, and N. Schmutz, "Extracting cloud motion vectors from satellite images for solar power forecasting," in *Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEEE International*. IEEE, 2014, pp. 4123–4126.

[11] R. Marquez and C. F. Coimbra, "Intra-hour dni forecasting based on cloud tracking image analysis," *Solar Energy*, vol. 91, pp. 327–336, 2013.

[12] H. S. Jang, K. Y. Bae, H.-S. Park, and D. K. Sung, "Solar power prediction based on satellite images and support vector machine," *IEEE Trans. Sustain. Energy*, vol. 7, no. 3, pp. 1255–1263, 2016.

[13] J. L. Bosch and J. Kleissl, "Cloud motion vectors from a network of ground sensors in a solar power plant," *Solar Energy*, vol. 95, pp. 13–20, 2013.

[14] K. Y. Bae, H. S. Jang, and D. K. Sung, "Hourly solar irradiance prediction based on support vector machine and its error analysis," *IEEE Transactions on Power Systems*, vol. 32, no. 2, pp. 935–945, 2017.

[15] C. Crisosto, M. Hofmann, R. Mubarak, and G. Seckmeyer, "One-hour prediction of the global solar irradiance from all-sky images using artificial neural networks," *Energies*, vol. 11, no. 11, p. 2906, 2018.

[16] M. Khodayar and M. Teshnehlab, "Robust deep neural network for wind speed prediction," in *Fuzzy and Intelligent Systems (CFIS), 2015 4th Iranian Joint Congress on*. IEEE, 2015, pp. 1–5.

[17] P. Lauret, C. Voyant, T. Soubdhan, M. David, and P. Poggi, "A benchmarking of machine learning techniques for solar radiation forecasting in an insular context," *Solar Energy*, vol. 112, pp. 446–457, 2015.

[18] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong, "Probabilistic forecasting of wind power generation using extreme learning machine," *IEEE Transactions on Power Systems*, vol. 29, no. 3, pp. 1033–1044, 2014.

[19] H. Le Cadre, I. Aravena, and A. Papavasiliou, "Solar pv power forecasting using extreme learning machine and information fusion," in *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2015, pp. 1–6.

[20] Y. Zhang and J. Wang, "Gefcom2014 probabilistic solar power forecasting based on k-nearest neighbor and kernel density estimator," in *Power & Energy Society General Meeting, 2015 IEEE*. IEEE, 2015, pp. 1–5.

[21] D. W. Van der Meer, J. Widén, and J. Munkhammar, "Review on probabilistic forecasting of photovoltaic power production and electricity consumption," *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 1484–1512, 2018.

[22] Y. Chu and C. F. Coimbra, "Short-term probabilistic forecasts for direct normal irradiance," *Renewable Energy*, vol. 101, pp. 526–536, 2017.

[23] J. Huang and M. Perry, "A semi-empirical approach using gradient boosting and k-nearest neighbors regression for gefcom2014 probabilistic solar power forecasting," *International Journal of Forecasting*, vol. 32, no. 3, pp. 1081–1086, 2016.

[24] F. Golestaneh, P. Pinson, and H. B. Gooi, "Very short-term nonparametric probabilistic forecasting of renewable energy generation—with application to solar energy," *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 3850–3863, 2016.

[25] C. Wan, J. Lin, Y. Song, Z. Xu, and G. Yang, "Probabilistic forecasting of photovoltaic generation: An efficient statistical approach," *IEEE Transactions on Power Systems*, vol. 32, no. 3, pp. 2471–2472, 2017.

[26] P. Lauret, M. David, and H. T. Pedro, "Probabilistic solar forecasting using quantile regression models," *Energies*, vol. 10, no. 10, p. 1591, 2017.

[27] F. Golestaneh, H. B. Gooi, and P. Pinson, "Generation and evaluation of space–time trajectories of photovoltaic power," *Applied energy*, vol. 176, pp. 80–91, 2016.

[28] R. Juban, H. Ohlsson, M. Maasoumy, L. Poirier, and J. Z. Kolter, "A multiple quantile regression approach to the wind, solar, and price tracks of gefcom2014," *International Journal of Forecasting*, vol. 32, no. 3, pp. 1094–1102, 2016.

[29] K. B. Debnath and M. Mourshed, "Forecasting methods in energy planning models," *Renewable and Sustainable Energy Reviews*, vol. 88, pp. 297–325, 2018.

[30] M. Khodayar, J. Wang, and M. Manthouri, "Interval deep generative neural network for wind speed forecasting," *IEEE Transactions on Smart Grid*, 2018.

[31] S. Chen, H. Gooi, and M. Wang, "Solar radiation forecast based on fuzzy logic and neural networks," *Renewable Energy*, vol. 60, pp. 195–201, 2013.

[32] P. Luo, S. Zhu, L. Han, and Q. Chen, "Short-term photovoltaic generation forecasting based on similar day selection and extreme learning machine," in *Power & Energy Society General Meeting, 2017 IEEE*. IEEE, 2017, pp. 1–5.

[33] A. Bracale, G. Carpinelli, P. De Falco, R. Rizzo, and A. Russo, "New advanced method and cost-based indices applied to probabilistic forecasting of photovoltaic generation," *Journal of Renewable and Sustainable Energy*, vol. 8, no. 2, p. 023505, 2016.

[34] R. Bayindir, M. Yesilbudak, M. Colak, and N. Genc, "A novel application of naive bayes classifier in photovoltaic energy prediction," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec 2017, pp. 523–527.

[35] A. Ciaramella, A. Staiano, G. Cervone, and S. Alessandrini, "A bayesian-based neural network model for solar photovoltaic power forecasting," in *International Workshop on Neural Networks*. Springer, 2015, pp. 169–177.

[36] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, and R. J. Hyndman, "Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond," *International Journal of Forecasting*, vol. 32, no. 3, pp. 896–913, 2016.

[37] J. Wang, P. Li, R. Ran, Y. Che, and Y. Zhou, "A short-term photovoltaic power prediction model based on the gradient boost decision tree." *Applied Sciences*, vol. 8, no. 5, pp. 2076–3417, 2018.

[38] W. Zhang, H. Quan, O. Gandhi, C. D. Rodríguez-Gallegos, A. Sharma, and D. Srinivasan, "An ensemble machine learning based approach for constructing probabilistic pv generation forecasting," in *Asia-Pacific Power and Energy Engineering Conference (APPEEC), 2017 IEEE PES*. IEEE, 2017, pp. 1–6.

[39] S. Tiwari, R. Sabzchgar, and M. Rasouli, "Short term solar irradiance forecast using numerical weather prediction (nwp) with gradient boost regression," in *2018 9th IEEE International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*. IEEE, 2018, pp. 1–8.

[40] A. Bracale, G. Carpinelli, and P. De Falco, "A probabilistic competitive ensemble method for short-term photovoltaic power forecasting," *IEEE Transactions on Sustainable Energy*, vol. 8, no. 2, pp. 551–560, 2017.

[41] Y. Chen, Y. Wang, D. Kirschen, and B. Zhang, "Model-free renewable scenario generation using generative adversarial networks," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 3265–3275, 2018.

[42] M. Khodayar, J. Wang, and Z. Wang, "Energy disaggregation via deep temporal dictionary learning," *arXiv preprint arXiv:1809.03534*, 2018.

[43] J. Zhang, Y. Wang, M. Sun, N. Zhang, and C. Kang, "Constructing probabilistic load forecast from multiple point forecasts: A bootstrap based approach," in *2018 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia)*. IEEE, 2018, pp. 184–189.

[44] 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.

[45] J. Tastu, P. Pinson, and H. Madsen, "Space-time scenarios of wind power generation produced using a gaussian copula with parametrized precision matrix," *Tech. Univ. Denmark, Tech. Rep.*, 2013.

[46] X. G. Agoua, R. Girard, and G. Kariniotakis, "Probabilistic model for spatio-temporal photovoltaic power forecasting," *IEEE Transactions on Sustainable Energy*, pp. 1–1, 2018.

[47] A. Tascikaraoglu, B. M. Sanandaji, G. Chicco, V. Cocina, F. Spertino, O. Erdinc, N. G. Paterakis, and J. P. Catalao, "Compressive spatio-temporal forecasting of meteorological quantities and photovoltaic power," *IEEE Transactions on Sustainable Energy*, vol. 7, no. 3, pp. 1295–1305, 2016.

[48] Y. A. Awad, P. Koutrakis, B. A. Coull, and J. Schwartz, "A spatio-temporal prediction model based on support vector machine regression: Ambient black carbon in three new england states," *Environmental research*, vol. 159, pp. 427–434, 2017.

[49] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[50] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[51] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[52] M. Khodayar and J. Wang, "Spatio-temporal graph deep neural network for short-term wind speed forecasting," *IEEE Transactions on Sustainable Energy*, 2018.

[53] S. Alessandrini, L. Delle Monache, S. Sperati, and G. Cervone, "An analog ensemble for short-term probabilistic solar power forecast," *Applied energy*, vol. 157, pp. 95–110, 2015.

[54] Z. Szabó, "Information theoretical estimators toolbox," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 283–287, 2014.

[55] M. Bastian, S. Heymann, M. Jacomy, *et al.*, "Gephi: an open source software for exploring and manipulating networks." *Icwsm*, vol. 8, no. 2009, pp. 361–362, 2009.

[56] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.

[57] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: a system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.