2021 International Conference on Smart Energy Systems and Technologies (SEST) | 978-1-7281-7660-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/SEST50973.2021.9543326

Deep Convolutional Graph Rough Variational Auto-Encoder for Short-Term Photovoltaic Power Forecasting

Mohsen Saffari
Department of Computer Science
University of Tulsa
Tulsa, OK, USA
mos0968@utulsa.edu

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

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

Miadreza Shafie-khah
School of Technology and Innovations
University of Vaasa
Vaasa, Finland
mshafiek@univaasa.fi

João P. S. Catalão
INESCTEC
Faculty of Engineering of University of Porto
Porto, Portugal
catalao@fe.up.pt

***Abstract*—Photovoltaic (PV) power is considered as one of the most promising sustainable energy resources in recent years. However, the existing intermittency in the nature of solar energy is a significant problem for the optimization of smart grids. In this paper, to overcome PV generation uncertainty and provide an accurate spatio-temporal (ST) PV forecast, we propose a novel deep generative convolutional graph rough variational autoencoder (CGRVAE) that captures each PV site's probability distribution functions (PDFs) of future PV generation in a modeled weighted graph. Having the learned PDFs enables CGRVAE to accurately generate the future values of PV power time series. To train and evaluate our model, we used the measurements of a set of PV sites in California, US. The sites are modeled as a weighted graph where each node represents PV measurements at each site while edges reflect their correlations. Using graph spectral convolutions the proposed model extracts the most relevant information of the graph to estimate the future PV given the historical time series for each node in the modeled graph. Experimental results show the superiority of CGRVAE over state-of-the-art forecasting approaches in terms of the root mean square error (RMSE) and mean absolute error (MAE) metric.**

***Index Terms*—Solar energy, Photovoltaic power forecasting, Spectral Graph Convolution, Deep Neural Networks**

NOMENCLATURE

α_j	jth Chebyshev coefficient
$\hat{s}_{n_i}$	Forecasted values of $s^*_{n_i}$
$\mathcal{F}_\theta$	Graph convolution filter
L_G	Set of edges in G
N_G	Set of nodes in G
P^*	PDF of future value of time series
$s^*_{n_i}$	Future values of $n_i th$ time series
$s^t_{n_i}$	$n_i th$ time series at time t
D	Decoder in VAE architecture
E	Encoder in VAE architecture
G	Weighted Indirected Graph

I. INTRODUCTION

SOLAR energy is one of the sustainable, clean, and free cost sources of energy. Nowadays, numerous large-scale solar photovoltaic (PV) systems are commonly employed in power systems of modern countries. The dependency of solar on different atmospheric parameters such as sunshine and temperature makes the PV power output (PVPO) intermittent and unpredictable. When large-scale power plants are connected to the power grid, the power instability of the PV systems is a severe threat to the power grid's safety and stability; hence, there is a must to propose an accurate and reliable PVPO forecasting method to improve the power plant stability [1].

Generally, the PVPO forecasting approaches can be classified into three categories including, 1) Statistical, 2) Machine learning (ML), 3) Hybrid methods. Statistical methods aim to forecast the future values by merely considering the previous values of the PV time series. Persistence models [2], auto-regressive integrated moving average (ARIMA) [3], and auto-regressive moving average model with exogenous inputs (ARMAX) [4] are among most common statistical methods.

ML-based methods are data-driven approaches that are exploited for a variety of regression [5] and classification [6] tasks. Recently ML models have been witnessed significant attention for the problem of PVPO forecasting. In [7] the authors applied the Long-Short Term Memory (LSTM) model to forecast the $\{6, 12, 24\}$ hours ahead of PVPO time series. To improve the forecasting ability of the LSTM model, Zhou et al. [8] adopted an attention mechanism in two LSTM networks to adaptively focus on more relevant input features for temperature and power outputs prediction. In [9], a feature vector selection (FVS) method is proposed to choose the most relevant solar data from high dimensional input space; moreover, a novel kernel extreme learning machine (KELM) model is developed for the forecasting task. By comparing the FVS-KELM with other ML baselines such as ELM and support vector machine,

978-1-7281-7660-4/21/$31.00 ©2021 IEEE

the authors empirically show the high generalization capacity of the proposed model. In [10], Yao et al. evaluate the validity of echo state network (ESN) on the problem of PVPO forecasting. In their proposed model, first, the relevant features of time series are extracted through unsupervised learning algorithms of Restricted Boltzmann Machine and principal component analysis, then inspiring by the concept of deep structures a multiple reservoirs ESN is developed for the prediction task. Recently, spatio-temporal (ST) approaches attract more interest for forecasting models. In [11], proposed QR-Lasso model which provides the complete future probability density function of PV production for very short-term horizons based on quantile regression and an L1 penalization technique for automatic selection of the input variables. Chai et al. [12] exploited ConvLSTM for PVOP forecasting problem. ConvLSTM is a variant of the LSTM model where all the inputs, cell outputs, hidden states, and gates are 3D tensors (instead of 2D matrices in original LSTM).

By combining different forecasting techniques, hybrid methods aim to propose a more efficient and accurate prediction [13]–[15]. For example, recently [16] proposed model consists of wavelet packet decomposition to decompose the original power time series into sub-series and four independent LSTM models to predict one hour-ahead PVPO values.

In this paper, we solved the problem of PVPO forecasting for a graph of PV sites by proposing a novel convolutional graph rough variational auto-encoder (CGRVAE) model. Unlike other studies in the literature that try to predict the PVPO time series of one PV site, here we propose a model that is able to forecast the time series for a graph of neighboring sites to highlight the ST correlations exist in solar generation. In this study, we model 25 PV sites in California, US, as an indirected weighted graph where the nodes represent the PV measurements at the sites while edges reflect their historical correlations. By making use of a convolutional graph, we extract the most relevant set of ST features within the graph. Having these extracted features and future time series values in the training set enables us to model the probability density function (PDF) of future PVPO given the historical observation for each site. To obtain a robust PDF approximator against the existing uncertainties in PV time series, we propose a novel rough variational autoencoder (RVAE) that approximates the PDF for each node, i.e., PV site, in the graph. By observing historical records of graph-structured information and learning the distribution for each node in the graph, the proposed deep CGRVAE has the capacity to generate new samples of future PV generation for each node in the modeled graph.

The main contributions of this study can be summarized as follows: 1) In contrast to the existing methods that merely employ discriminative learning to directly map the history of PV generation to future values, here we learn a complex PDF of future solar generation given the history and take samples to forecast PV generation. The proposed generative model is shown to obtain higher accuracy compared to the existing works; 2) Rough set theory is incorporated into the proposed deep generative model to handle noise and uncertainties in the ST PV forecasting datasets; and 3) To the best of our knowledge, this is the first attempt to learn conditional ST PDFs using a deep learning data-driven algorithm. The proposed model not only provides higher generalization capacity compared to recent benchmarks, but also takes into account the robust feature extraction from PV datasets.

The paper is organized as follows: In Section, II the underlying problem and modeled weighted graph for this study is defined. Section III discuss mathematical modeling of the proposed forecasting methodology. Sections IV explains the used performance metrics and the obtained results. Finally, conclusion are presented in Section V.

II. Problem Formulation

The PVPO time series of 25 sites in 2006 in California are collected by National Solar Radiation Database [17]. Fig. 1 illustrates the location of the sites. The data at each site contains the PVPO time series with 5-min intervals. Let us define the weighted indirected PV graph, $G = \langle N_G, L_G \rangle$ in which N_G consists of $n_i,\ i \in \{1, 2, ..., 25\}$ nodes. i.e., PV sites, and L_G is the set of edges between nodes in the graph. The weight matrix of the edges, W is obtained by the existing mutual information (MI) and geographical distance among nodes in the graph. The MI value of historical PV data for each pair of sites has large negative correlation with their distance inside the latitude-longitude space. It means that the shorter distances between solar sites, the larger values of edges for corresponding nodes in the graph. More formally, we define the i-th and j-th element of the weight matrix, W, by:

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

D_{ij} is the geographical distance between nodes i and j, while $MI(i,j)$ represents the normalized MI between two underlying nodes. The sparsity of the graph is determined by edge-sparsity coefficient, λ. The higher value of λ, the sparser weighted graph. In this work we set the edge sparsity value $\lambda = 0.5$ that is obtained by a validation set. For the node i at time t we have $S^t_{n_i}$ as the corresponding PVPO time series. Here the problem is to estimate the future values of the time series $S^*(t^{'} = t + h)$ for a considered forecasting horizon $h > 0$. To forecast these values for all of the nodes in the graph G we need to learn a conditional PDF, $P^* \left(S^*(t^{'}) | \psi \right)$ where ψ is the history of PV generation at all nodes. After learning P^*, one can generate future values of the PV at each site.

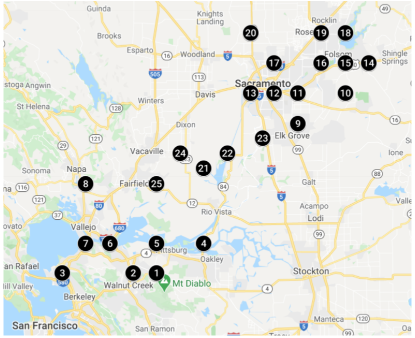

Fig. 1: Locations of 25 underlying PV sites in California, USA.

Fig. 2 shows the MI analysis between the PVPO values of site 2 at time $\widetilde{t}$ and $\widetilde{t} - k$ with $k \in \{12, 24, 36, ..., 144\}$ hours. We found $s(\widetilde{t})$ is more correlated with most recent PV values as well as the values in range $\{24, 48, 72, 96, 120, 144\}$ hours.

Hence, in this work to predict the $s(\widetilde{t})$ we feed the historical PV values with $MI > 0.4$ to the proposed model.

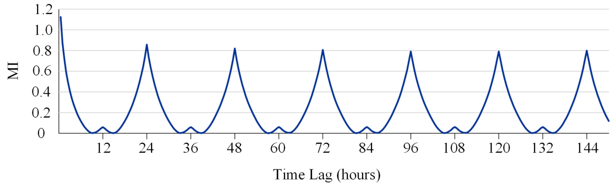

Fig. 2: Correlation between current time and previous time lags of the second solar site

III. Proposed model

A. Graph Convolutional Neural Network

As discussed in Section II, we aim to learn the PDF of the PVPO time series to predict the future values of the time series. As shown in Fig. 3 to obtain the relevant ST features ($F(G)$) of the nodes in the graph, we used graph convolutional networks [18]. At the time t, the spectral graph convolution of G is computed by:

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

where S^t represents all nodes in the graph G at time t. Matrix V contains the eigenvector of the normalized Laplacian $L = V\Omega V^t$ and the vector $\theta \in R^n$ is the set of parameters for convolutional filter $\mathcal{F}_\theta = diag(\theta)$ in the frequency domain. Since $\mathcal{F}_\theta$ is considered as function of the eigenvectors for matrix L, the filter in the model is denoted by $\mathcal{F}_\theta(\Omega)$. By estimation of $\mathcal{F}_\theta(\Omega)$ using Chebyshev Polynomials, T_j, we have $\mathcal{F}_\theta(\Omega) \approx \sum_{j=0}^{J} \alpha_j T_j \left(\frac{2}{\vartheta_{max}}\Lambda - I\right)$ where α_j is the j-th Chebyshev coefficient and ϑ_{max} is the maximum eigen value of L. Substituting this approaximation of $\mathcal{F}_\alpha(\Omega)$ into (2) gives us a new expression for spectral graph convolution function of G:

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

The convolution equation in (3) could be more simplified by supposing $J = 1$, $\vartheta_{max} = 2$ and $\alpha_0 = -\alpha_1$. The simplified version of (3) is:

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

To extract the ST features of the nodes in graph G using the convolutional operation of (4), the proposed model shown in Fig. 3 considers L_G layers in the Graph feature extractor (GFE) block. The output of the k-th layer in GEF, O^k, is obtained by:

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

where W is the graph weight matrix defined in (1), β^k is the weights for k-th layer in neural network (NN), and $\tilde{B}_{ij} = \sum_j \left(W + I\right)_{ij}$. As shown in the Fig. 3, the input to the GEF network is the raw data of the historical time series, and its output $F(G)$ is the ST features of graph G.

B. Time Series Approximation by PDF Learning

In Section III-A, we discuss ST feature extraction of graph G. In this Section, our objective is to learn the $P(X)$ over high dimensional data points $X \in \mathcal{X}$. Then, in Section III-B2 we extend the math to our problem, learning the $P^*\left(S^*|\psi\right)$. Having P^* we can generate the future values of time series, S^*, as close as possible to the observed samples, S, in training phase.

1) Learning Probabilistic Representation of the Data: As the input space's complexity $\mathcal{X}$ grows, the difficulty of accurate $P(X)$ approximation grows. Therefore, we map the inputs into a latent random space, $\mathcal{Z}$, to represent the most important characteristic of $P(X)$. Here we need to ensure that by sampling from unknown distribution $P(z)$ over high dimensional space $\mathcal{Z}$, we are able to generate some samples $\hat{X}$ that follow the original PDF $P(X)$. Suppose a set of deterministic functions $f(z;\theta)$ with parameters $\theta \in \Theta$ for mapping data points from $\mathcal{Z}$ space to $\mathcal{X}$ space, i.e., $f : \mathcal{Z} \times \Theta \to \mathcal{X}$. Our objective is to find a set of optimal parameters $\theta^* \in \Theta$ for f such that when $z \sim P(z)$ the probability of generating samples X^* as close as possible to X by f is maximized. Therefore, the optimization problem is written by:

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

Since $\mathcal{Z}$ is a transformed space of the input space $\mathcal{X}$, for a fixed set of parameters θ, $f(z;\theta)$ is a random vector in the space $\mathcal{X}$. Hence, $P(X)$ in (6) can be written as:

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

where D is decoder NN in variational auto-encoder (VAE) model. We consider conditional PDF $D\left(X|z;\theta\right)$ and prior PDF, $P(z)$, as a Gaussian distributions, $\mathcal{N}(X|f(z;\theta), \sigma^2 \times I)$ and $\mathcal{N}(0, I)$, respectively. First layer of decoder network maps variables $z \in \mathcal{Z}$ into an unknown complicated distributions ζ via nonlinear neurons, and afterwards ζ provides samples $X \in \mathcal{X}$. Considering the explained assumptions let us rewrite the optimization problem in (6),(7) by:

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

To solve the optimization problem in (8) we need to decide on feasibility of an arbitrary $z \in \mathcal{Z}$ in the generation of new samples belonging to $\mathcal{X}$. Hence, we define an arbitary probability distribution $E(z)$. The expected value of $D(X|z)$ w.r.t. z, $\mathbb{E}_{z\sim E}\left[D\left(X|z\right)\right]$, is obtained by Kullback–Leibler (KL) divergence.

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

By applying Bayes rules on $D\left(z|X\right)$ we can write the KL divergence as:

$$\begin{aligned} &\mathrm{KL}[E\left(z\right), P\left(z|X\right)] = \mathbb{E}_{z\sim E}\left[\log E\left(z\right) - \log\left(\tfrac{P(X|z)P(z)}{P(X)}\right)\right] = \\ &\mathrm{E}_{z\sim E}\left[\log E\left(z\right) - \log P\left(X|z\right) - \log P\left(z\right) + \log P\left(X\right)\right] \end{aligned} \tag{10}$$

where $P(X|z)$ is a decoder NN; therefore, we denote it by $D(X|z)$. Also, since we aim to infer $P(X)$, the distribution E has to be dependent on X. The above equality can be written as:

$$\begin{aligned} &\log P\left(X\right) - KL\left[E\left(z|X\right)\|P\left(z|X\right)\right] = \\ &\qquad \mathbb{E}_{z\sim E}\left[\log D\left(X|z\right) - KL\left[E\left(z|X\right)\|P\left(z\right)\right]\right] \end{aligned} \tag{11}$$

To achieve our goal, i.e., generating $X^* \approx X$, we must maximize $\log\left(P\left(X\right)\right)$ as well as minimize $KL\left[E\left(z|X\right)\|P\left(z|X\right)\right]$ in left hand side of (11); thus, we maximize the right hand side of (11) using stochastic gradient descent (SGD) method. Formally we can write the optimization problem by:

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

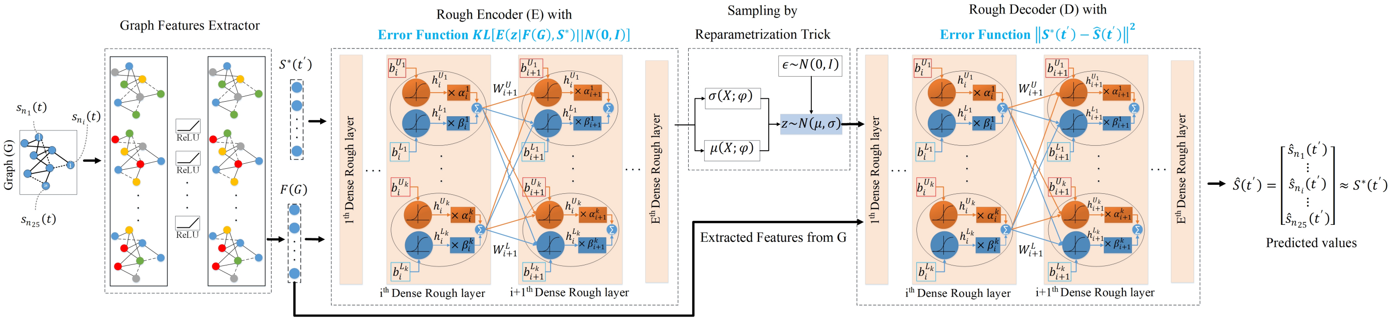

Fig. 3: Proposed Model CGRVAE on training phase. The trained rough decoder is used on testing phase.

Note that in (11), E is an encoder NN to encode the input samples X into z, and D is a decoder NN to X from z. We define E as:

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

where μ and Σ are the deterministic functions obtained by a NN with tuanable parameters θ. Since D and P both are multivariate Guassian distributions, the KL term in (11) can be simplified by:

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

Using reparametrization trick on z for obtaining $D\left(X|z;\theta\right)$ (see Fig. 3), (14) can be rewritten by:

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

In training phase, the encoder NN, E, receives the input data, and outputs μ and Σ (see (13)). The error function for E NN is computed in (14). After obtaining μ and Σ, by applying reparametrization trick, we can obtain z and feed that to decoder network D to generate $X^* \approx X$. Note that the error function for D NN is $||X - X^*||^2$.

2) *Convolutional Graph Rough VAE:* In this section, we extend the model in III-B1 for learning $P^*(S^*|\psi)$. As shown in Fig. 3 our goal is to generate $\hat{S}(t') \approx S^*$, hence let us formalize the generation steps of $\hat{S}$ as an approximation of S^*. Assuming $z \sim E$, Bayes rule is employed to obtain the expected value of $\log[P(S^*(t')|z,\psi)]$:

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

(16) can be rewritten as:

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

Similar to (11) our objective is to maximize the left hand side of (17). Hence, by solving similar optimization problem in (12) we learn the optimal Rough encoder/decoder to capture the conditional PDF $P^*(S^*|\psi)$. Having the optimal encoder/decoder enables us to generate the accurate values of S^* for future values t'. As shown in Fig. 3 and similar to (14) the cost function for tuning the encoder and decoder models in CGRVAE are defined as follows,

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

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

As shown in Fig. (3) in the proposed CGRVAE, we use Rough encoder/decoder NNs. The Rough neurons in the model include nonlinear activation functions with interval weights and biases. There are L_D and L_E Rough layers in the encoder and decoder networks, respectively. Here, we merely write the feedforward of the used Rough network between two layers i and $i+1$ by:

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

where $\langle W_{i+1}^U, b_{i+1}^U\rangle$ and $\langle W_{i+1}^L, b_{i+1}^L\rangle$ are upper bound and lower bound (i.e., interval weights) of the Rough NNs, respectively. Our main motivation to making use of Rough set theory and Rough neurons [19] in encoder/decoder models is to boost the robustness of model against the existing uncertainties in the PVPO data.

IV. Numerical Results

A. Experimental Details

CGRVAE is compared with several recent approaches for PV power forecasting, including Quantile Regression (QR) [20], Feature Vectors Selection Kernel Extreme Learning Machines (FVS-KELM) [9], Probabilistic Persistence (PP) [21], Long-Short Term Memory (LSTM) [7], attention mechanism with multiple LSTM (ALSTM) [8], multiple reservoirs echo state network (MR-ESN) [10], and two ST approaches, Convolutional LSTM (ConvLSTM) [12] and quantile regression (QR-Lasso) [11]. We compare the proposed model with other baselines over five forecasting horizons {10-min,30-min,1-hour,3-hours,6-hours} using RMSE and MAE metrics that are defined as follows:

$$\begin{aligned}RMSE &= \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(S^*\left(n\right) - \hat{S}\left(n\right)\right)^2} \\ MAE &= \frac{1}{N}\sum_{n=1}^{N}\left|S^*\left(n\right) - \hat{S}\left(n\right)\right|\end{aligned} \qquad (20)$$

where N is the number of test samples while the actual and predicted values of time series are shown by S^* and $\hat{S}$, respectively.

In this work, we split the dataset by seasons. For each season we consider 80% of samples as training and validation sets as well as 20% as testing set. The entire seasonal training and testing sets are contains 21024 and 5256 samples. Stochastic Gradient Descent with learning rate $\eta = 5 \times 10^{-4}$ is employed to train the CGRVAE. The number of layers for GFE, E, and D models are set as $L_{DFE} = 2$, $L_E = 4$, $L_D = 3$, respectively. All the experiments are carried out using GPU-based Tensorflow on Python 3, and the whole training time for our model is 12.3 minutes. The simulations are processed in a system with a 10-core CPU having Intel core-i7 Processors, an NVidia Quadro RTX 6000 GPU, and a 256-GB RAM.

B. Performance Comparison

Fig. 4 compares the 1 day prediction results of the proposed model and the baselines for 1-hour forecasting horizon of 14th solar site. In this figure, we can see the forecasting curve of the proposed CGRVAE is closer to the actual curve; however, there is a gap for other baselines (i.e., MR-ESN, QR-Lasso, and ConvLSTM).

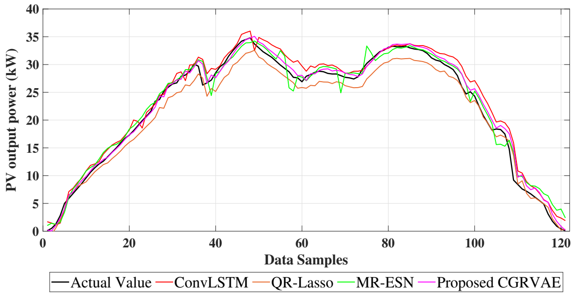

Fig. 4: One-hour ahead forecasting results of PVPO of 14th PV site. Note that for the sake of visualization, merely the results are shown for one day in spring and the zero value are not shown.

Comprehensive comparisons of the models' performance are presented in Tables I, II. The tables compare the RMSE and MAE of the proposed model with baselines in 10-mins, up to 6-hours ahead forecasting horizons in 4 seasons. Based on the tables, the RMSE and MAE metrics are rising monotonically by increasing the forecasting horizons. Moreover, to better visualize the results in tables, in Fig. 5 we compare the average RMSE of different methods for underlying forecasting time horizons. The average RMSE of PP is around 0.9 kW in 10-min, which is increased to 2.5 kW in 6-hours ahead forecast. In line with [12], our results confirm that statistical approaches (i.e., PP and QR) model yield higher values in RMSE and MAE due to their smoothness assumption. This assumption degrades the PP model's performance, especially for the days with many changes between cloudy and sunny weather. Generally, ML-based approaches outperform statistical methods, especially for larger forecasting horizons. For example, as shown in 5, the average RMSE of FVS-KELM and QR is almost the same in 10-mins; however, in a 3-hour forecast, FVS-KELM and QR yield 1.86 and 2.1 average RMSE, respectively.

Recurrent structures (i.e., LSTM and ALSTM) improves the forecasting results of QR and FVS-KELM; for example, LSTM outperforms QR and FVS-KELM with 16% and 9% average RMSE improvement in 6-hours forecasts, respectively. This superiority is because recurrent approaches can better capture the temporal characteristics of time series. In line with [8], our observation in tables I and II show the improvement in LSTM's results by applying the attention mechanism. The obtained results for the MR-ESN method verify the superiority of Echo State Networks on recurrent NN [10]. MR-ESN achieves 0.5 for the average RMSE in the 10-min forecast, while ALSTM yields a higher value around 0.65. The gap between these two models increases for larger forecast horizons. For example, in the 6-hours forecast, the average RMSE for MR-ESN and ALSTM are around 1.78 and 1.94, respectively.

One can observe the lower values of average RMSE for ST approaches (i.e., QR-Lasso, ConvLSTM, and proposed CGRVAE) compared to other referred benchmarks on Fig. 5. As concluded in [11], [12], our results prove the importance of considering spatial correlations as well as a temporal correlation for PV forecasting task. Based on the forecasting results on Fig. 4 and 5, we see ConvLSTM yields slightly better results compare QR-Lasso. For example, on 3 hours forecasting horizons, ConvLSTM merely improved the average RMSE by around 0.1. Obtained results show the graph-based NN and interval VAE caused the higher generalization capacity of the proposed CGRVAE method among all statistical and ML-based (temporal and ST) baselines. As shown in Fig. 5, the proposed model outperforms MR-ESN (as the best temporal ML-based) and ConvLSTM (as the best ST ML-based) by 9% and 3% on 6-hours forecasting horizons, respectively. These improvements in the results stem from the higher capacity of graph structures in modeling ST information and the superiority of generative modeling over discriminative modeling.

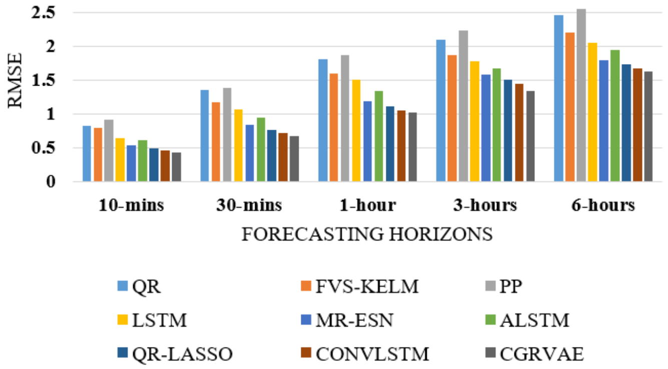

Fig. 5: Overall RMSE of different models for five forecasting horizons.

V. Conclusion

This paper presents a novel deep generative model, convolutional graph rough variational autoencoder, to learn the probability densities of a graph of photovoltaic sites. Here, we formulate the PV power forecasting problem as a graph distribution learning where each PV site is modeled as a node in the graph, and the distance between sites is modeled by edges in the graph. The model applies a deep convolutional graph to extract spatial features between the nodes in the graph. Further, the extracted features are fed to a rough variational auto-encoder to estimate the probability density of future PV power outputs. Numerical results show the remarkable superiority of the proposed model compared to state-of-the-art forecasting benchmarks. Compared to the method with a lack of spatio-temporal (ST) information (e.g., ALSTM), the proposed CGRVAE significantly boosts the

TABLE I: RMSE(kW) of different models for five forecast horizons on the testing datasets across seasons. Note that the best test results are marked in **bold** fonts.

Model	10-min				30-min				1-hour				3-hours				6-hours			
	Summer	Autumn	Winter	Spring	Summer	Autumn	Winter	Spring	Summer	Autumn	Winter	Spring	Summer	Autumn	Winter	Spring	Summer	Autumn	Winter	Spring
QR [20]	0.727	0.846	0.891	0.823	1.131	1.598	1.364	1.323	1.639	1.893	1.956	1.725	1.982	2.021	2.198	2.159	2.312	2.438	2.502	2.564
FVS-KELM [9]	0.674	0.832	0.929	0.758	1.023	1.236	1.321	1.112	1.569	1.532	1.674	1.614	1.823	2.012	1.997	1.846	2.193	2.368	2.422	2.523
PP [21]	0.895	0.834	0.984	0.964	1.242	1.452	1.426	1.387	1.701	1.921	1.825	2.006	2.128	2.333	2.196	2.269	2.441	2.503	2.691	2.552
LSTM [7]	0.594	0.651	0.689	0.649	0.985	0.921	1.150	1.212	1.449	1.323	1.578	1.652	1.774	1.795	1.802	1.715	1.951	2.015	2.110	2.106
MR-ESN [10]	0.516	0.502	0.610	0.536	0.798	0.842	0.892	0.833	1.129	1.143	1.192	1.259	1.584	1.458	1.622	1.635	1.736	1.726	1.836	1.842
ALSTM [8]	0.530	0.578	0.674	0.6333	0.846	0.894	0.919	1.113	1.236	1.299	1.397	1.423	1.618	1.678	1.666	1.701	1.849	1.972	2.032	1.911
QR-Lasso [11]	0.478	0.482	0.495	0.471	0.753	0.766	0.759	0.747	1.002	1.11	1.111	1.198	1.498	1.410	1.522	1.578	1.679	1.697	1.754	1.771
ConvLSTM [12]	0.443	0.453	0.461	0.455	0.692	0.710	0.733	0.701	0.971	0.989	1.089	1.151	1.401	1.366	1.475	1.498	1.601	1.645	1.702	1.755
Proposed CGRVAE	**0.402**	**0.419**	**0.443**	**0.425**	**0.618**	**0.678**	**0.721**	**0.654**	**0.942**	**0.984**	**1.012**	**1.109**	**1.352**	**1.289**	**1.312**	**1.374**	**1.576**	**1.603**	**1.656**	**1.682**

TABLE II: MAE (kW) of different models for five forecast horizons on the testing datasets across seasons. Note that the best test results are marked in **bold** fonts.

Model	10-min				30-min				1-hour				3-hours				6-hours			
	Summer	Autumn	Winter	Spring	Summer	Autumn	Winter	Spring	Summer	Autumn	Winter	Spring	Summer	Autumn	Winter	Spring	Summer	Autumn	Winter	Spring
QR [20]	0.633	0.759	0.761	0.721	0.984	1.258	1.234	1.147	1.498	1.622	1.846	1.629	1.785	1.982	2.098	2.009	2.249	2.299	2.384	2.495
FVS-KELM [9]	0.598	0.692	0.784	0.582	0.918	1.055	1.138	0.946	1.459	1.328	1.512	1.486	1.626	1.965	1.733	1.628	1.941	2.269	2.341	2.369
PP [21]	0.759	0.729	0.798	0.859	1.004	1.239	1.249	1.189	1.514	1.721	1.543	1.863	1.936	2.128	2.009	2.184	2.259	2.325	2.583	2.389
LSTM [7]	0.442	0.478	0.514	0.549	0.786	0.741	0.982	1.023	1.249	1.148	1.328	1.419	1.581	1.623	1.624	1.541	1.802	1.92	2.039	1.984
MR-ESN [10]	0.359	0.339	0.469	0.398	0.541	0.652	0.712	0.612	0.984	0.978	1.001	1.114	1.367	1.278	1.425	1.458	1.617	1.584	1.649	1.698
ALSTM [8]	0.361	0.435	0.498	0.512	0.65	0.69	0.754	0.961	1.011	1.131	1.152	1.254	1.412	1.503	1.485	1.511	1.632	1.789	1.95	1.769
QR-Lasso [11]	0.344	0.329	0.426	0.361	0.511	0.591	0.677	0.589	0.892	0.911	1.056	1.002	1.297	1.195	1.374	1.380	1.545	1.510	1.589	1.610
ConvLSTM [12]	0.303	0.321	0.409	0.343	0.503	0.556	0.626	0.561	0.821	0.884	0.988	0.994	1.226	1.110	1.333	1.302	1.510	1.523	1.540	1.597
Proposed CGRVAE	**0.298**	**0.303**	**0.312**	**0.308**	**0.410**	**0.523**	**0.535**	**0.489**	**0.739**	**0.787**	**0.891**	**0.926**	**1.149**	**1.066**	**1.132**	**1.187**	**1.322**	**1.455**	**1.478**	**1.532**

forecasting RMSE (0.334 in 3 hours forecasting horizons). This superiority reveals the importance of ST correlations between PV sites. Moreover, in comparison with ST approaches such as QR-Lasso and ConvLSTM, the proposed CGRVAE model improved the forecasting RMSE by 5.1% and 3% in 6 hours forecasting horizons. Having the accurate probability density of the future values of the time series enables us to predict the future values with lower error than most recent discriminative models.

References

[1] U. K. Das, K. S. Tey, M. Seyedmahmoudian, S. Mekhilef, M. Y. I. Idris, W. Van Deventer, B. Horan, and A. Stojcevski, "Forecasting of photovoltaic power generation and model optimization: A review," *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 912–928, 2018.

[2] M. Lipperheide, J. Bosch, and J. Kleissl, "Embedded nowcasting method using cloud speed persistence for a photovoltaic power plant," *Solar Energy*, vol. 112, pp. 232–238, 2015.

[3] H. T. Pedro and C. F. Coimbra, "Assessment of forecasting techniques for solar power production with no exogenous inputs," *Solar Energy*, vol. 86, no. 7, pp. 2017–2028, 2012.

[4] Y. Li, Y. Su, and L. Shu, "An armax model for forecasting the power output of a grid connected photovoltaic system," *Renewable Energy*, vol. 66, pp. 78–89, 2014.

[5] A. Noormohammadi-Asl, M. Saffari, and M. Teshnehlab, "Neural control of mobile robot motion based on feedback error learning and mimetic structure," in *Electrical Engineering (ICEE), Iranian Conference on*. IEEE, 2018, pp. 778–783.

[6] M. Saffari, M. Khodayar, M. S. Ebrahimi Saadabadi, A. F. Sequeira, and J. S. Cardoso, "Maximum relevance minimum redundancy dropout with informative kernel determinantal point process," *Sensors*, vol. 21, no. 5, p. 1846, 2021.

[7] M. S. Hossain and H. Mahmood, "Short-term photovoltaic power forecasting using an lstm neural network and synthetic weather forecast," *IEEE Access*, vol. 8, pp. 172 524–172 533, 2020.

[8] H. Zhou, Y. Zhang, L. Yang, Q. Liu, K. Yan, and Y. Du, "Short-term photovoltaic power forecasting based on long short term memory neural network and attention mechanism," *IEEE Access*, vol. 7, pp. 78 063–78 074, 2019.

[9] J. Li, M. Li *et al.*, "Short-term photovoltaic power prediction based on fvs-kelm method," *Journal of Applied Science and Engineering*, vol. 23, no. 2, pp. 289–301, 2020.

[10] X. Yao, Z. Wang, and H. Zhang, "A novel photovoltaic power forecasting model based on echo state network," *Neurocomputing*, vol. 325, pp. 182–189, 2019.

[11] X. G. Agoua, R. Girard, and G. Kariniotakis, "Probabilistic models for spatio-temporal photovoltaic power forecasting," *IEEE Transactions on Sustainable Energy*, vol. 10, no. 2, pp. 780–789, 2018.

[12] S. Chai, Z. Xu, Y. Jia, and W. K. Wong, "A robust spatiotemporal forecasting framework for photovoltaic generation," *IEEE Transactions on Smart Grid*, vol. 11, no. 6, pp. 5370–5382, 2020.

[13] 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.

[14] 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.

[15] M. Khodayar, M. E. Khodayar, and S. M. J. Jalali, "Deep learning for pattern recognition of photovoltaic energy generation," *The Electricity Journal*, vol. 34, no. 1, p. 106882, 2021.

[16] P. Li, K. Zhou, X. Lu, and S. Yang, "A hybrid deep learning model for short-term pv power forecasting," *Applied Energy*, vol. 259, p. 114216, 2020.

[17] 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.

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

[19] M. Khodayar, O. Kaynak, and M. E. Khodayar, "Rough deep neural architecture for short-term wind speed forecasting," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 2770–2779, 2017.

[20] 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.

[21] 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.