Spatio-Temporal Denoising Graph Autoencoders with Data Augmentation for Photovoltaic Data Imputation

Yangxin Fan, Xuanji Yu, Raymond Wieser
Case Western Reserve University
{yxf451,xxy530,rxw497}@case.edu

David Meakin
SunPower Corporation
david.meakin@sunpowercorp.com

Avishai Shaton
SolarEdge Technologies
avishai.shaton@solaredge.com

Jean-Nicolas Jaubert
CSI Solar Co.Ltd.
Jn.jaubert@csisolar.com

Robert Flottemesch
Brookfield Renewable U.S.
robert.flottemesch@luminace.com

Michael Howell
C2 Energy Capital
mh@c2.energy

Jennifer Braid
Sandia National Labs
jlbraid@sandia.gov

Laura S.Bruckman, Roger H.French, Yinghui Wu
Case Western Reserve University
{lsh41,rxf131,yxw1650}@case.edu

ABSTRACT

The fast growth of the global Photovoltaic (PV) market enables large-scale PV data analytical pipelines for power forecasting and long-term reliability assessment of PV fleets. Nevertheless, the performance of PV data analysis heavily depends on the quality of PV timeseries data. This paper proposes a novel Spatio-Temporal Denoising Graph Autoencoder (STD-GAE) framework, to impute missing PV Power Data. STD-GAE exploits temporal correlation, spatial coherence and value dependencies from domain knowledge to recover missing data. It is empowered by two modules. (1) To cope with sparse yet various scenarios of missing data, STD-GAE incorporates a domain-knowledge aware data augmentation module that creates plausible variations of missing data patterns. This generalizes STD-GAE to robust imputation over different seasons and environment. (2) STD-GAE nontrivially integrates spatiotemporal graph convolution layers (to recover local missing data by observed "neighboring" PV plants) and denoising autoencoder (to recover corrupted data from augmented counterpart) to improve the accuracy of imputation accuracy at PV fleet level. Using large-scale real data over 98 PV systems, our experimental study shows that STD-GAE achieves a gain from 8.38% to 45.44% in accuracy (MAE), and remains less sensitive to missing rate, different seasons and missing scenarios, compared with state-of-the-art data imputation methods such as MIDA and LRTC-TNN.

1 INTRODUCTION

Photovoltaics have become a dominant force in the energy sector over the past 20 years. The total, installed solar capacity has increased 500 times since 2000 to a total of 773 GW at the end of 2020 [8]. The exponential growth of the PV market has pushed the demand for power forecasting and performance evaluation for a huge population of PV power plants which have spatiotemporal coherence that can be utilized for improving model accuracy [15]. However, real-time sensor measurements are prone to disruptions caused by measurement error, unexpected shutdowns of meters, equipment or component faults, power outrage, communication failures, etc [6]. These disruptions lead to single missing point and large blocks of missing data. The existence of missing data may impact the performance of downstream timeseries analyses such as the long-term photovoltaic degradation rate estimation [28].

Imputing missing timeseries data has been extensively studied. Notable examples include interpolation approaches [4, 20, 25], statistical learning [32, 33], or imputation with physical models [34]. Conventional PV data imputation is designed to impute the data for individual PV plant or inverter, which often assume a certain missing data distribution (e.g., missing at random), or require extra physical information to recovery missing data [34]. Deep models such as Graph Neural Networks (GNNs) have delivered promising results in predicting time-series data, such as traffic forecasting [10, 35–37]. While data imputation can be considered as a predictive task, existing models often assume high-quality, complete input data, a luxury that one does not have for real-world PV data.

There are several major challenges in PV data imputation:

- *Multiple missing scenarios*: the missingness of PV data may be characterized by different missing patterns, which are often hard to be captured and imputed by a single model;
- *Lack of high-quality input*: It is also hard to obtain complete input as well as sufficient training examples, especially when there are multiple missing patterns;
- *Seasonality*: even with full observation, the distribution of PV data may vary due to seasonal variants, geospatial locations and weather factors.

For example, given the raw PV data collected from 98 PV inverters (8 PV sites) that spans over three year (2014-2017), we found in total 27 different missing data patterns (determined by whether the value of a specific attribute is missing), as illustrated in Fig. 1. Among the top 13 most frequent patterns, the most frequent one indicates a "worst case" that all attribute values are missing. 0.9964% of the records have at least power value ("lacp") missing.

These call for effective PV data imputation framework that can perform *unsupervised* and accurate imputation, bearing *noisy and incomplete* input, and in the presence of *multiple* missing patterns.

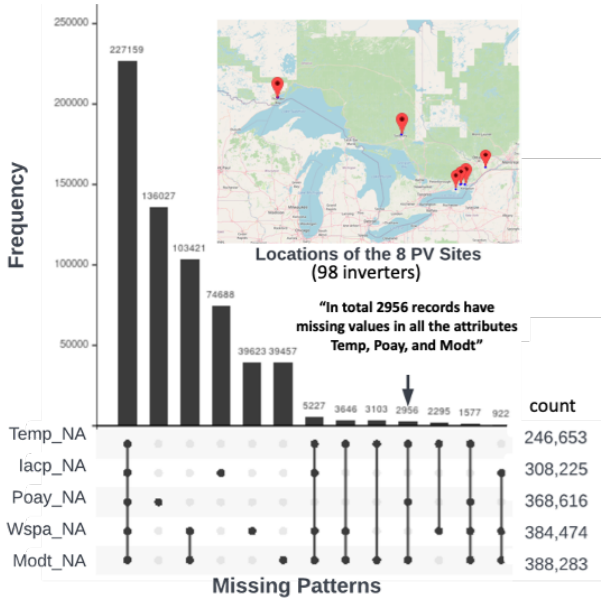


Figure 1: Illustration of most frequent (top 13 out of in total 27) missing data patterns and their frequency in a PV dataset of 98 PV inverters. Count: the number of records (*e.g.,* 246,653) having at least one (*e.g.,* temp) attribute value missing.

STD-GAE Framework. In response, we propose the *Spatio-Temporal Denoising Graph Autoencoder* (**STD-GAE**), a novel framework empowered by spatiotemporal GNNs to impute PV power data. Unlike conventional data imputation approach, STD-GAE is optimized for PV data imputation with the following characteristics.

(1) STD-GAE is empowered by a spatiotemporal graph autoencoder to accurately learn PV network representation for missing PV data imputation. Our intuition is that PV fleet can be modeled as a spatiotemporally correlated inverter network. The measurement from one inverter often help imputing its "similar" counterpart, captured by spatiotemporal correlations. To this end, STD-GAE adopts spatial, temporal, and inverter-level features in spatiotemporal graph convolution layers to learn PV fleet representations for imputation.

(2) To cope with sparse example, STD-GAE exploits a domain knowledge-aware data augmentation module. The module (a) leverages a suite of "plug-able" basic imputation methods to augment the input with imputed values, and (b) exploits a set of guard conditions from domain knowledge and physical models to validate the augmented input. This "cold-starts" STD-GAE learning with reasonable auxiliary data from even sparse PV observations.

(3) To generalize the imputation for PV input with multiple missing scenarios, STD-GAE takes a strategy of denoising autoencoders, whose goal is to learn accurate representation when part of input is missing, with an enhanced data corruption module, which allows configurable corruption with different missing types (*e.g.,* missing at random, block missing). By "enforcing" STD-GAE to reconstruct the corrupted yet augmented PV input, the imputation is able to achieve good performance for different scenarios.

Using real-world data collected from 98 PV inverters in Canada, we verified that STD-GAE can achieve a gain of 35.52% (resp. 15.09%) on average in MAE (resp. RMSE), compared to the state-of-the-art data imputation methods.The performance remains robust (not sensitive) even when the training of STD-GAE is constrained to a certain fraction of observations in a year.

Related Work. We summarize related work as follows.

PV data imputation. Conventional data-driven PV imputation usually adopt interpolation, statistical models or physical models. Notable examples include K-nearest Neighbor (KNN) and Linear Interpolation (LI), which interpolate missing data points by aggregating their neighboring ones[4, 20, 25]; and Multiple Imputation by Chained Equations (MICE) [32, 33], which uses statistical model and assumes missing at random pattern (MAR). These method only focus on imputing a single PV inverter or module, and often lead to biases if multiple missing patterns co-exists. Physical models have been proposed to use fully observed correlated attributes to recovery missing data in the target PV timeseries attribute [21, 34]. One of the major disadvantages is that if predictors are also missing, the model cannot sufficiently recover missing data. In addition, physical models rely on material and physical parameters that are highly variable and not well documented.

Spatiotemporal Graph Neural Networks (ST-GNNs). STGNNs extends GNNs to model spatiotemporal networks with *e.g.,* recurrent graph convolutions [3, 31] or attention aggregated layers [36, 38]. The former captures spatiotemporal coherence by filtering inputs and hidden states passed to a recurrent unit using graph convolutions, while the latter learns latent dynamic temporal or spatial dependency through convolutions or attention mechanisms. Compared to STGNNs based prediction [35], we use an expressive sandwiched Spatio-temporal block, utilize data augmentation to improve the input quality, and support configurable data corruption to cope with diverse missing scenarios. To the best of our knowledge, this is the first work that integrates denoising autoencoders and spatiotemporal graph autoencoders for PV data imputation.

Organization. The remainder of this paper is structured as follows. Section 2 presents a brief introduction to Graph Neural Networks and Autoencoders and formulates the imputation problem. Section 3 introduces the proposed modular imputation method and provide details about each component. Section 4 represents and analyzes experimental results. Section 5 proposes application deployment of our model. Finally, the paper is concluded in Section 6.

2 PROBLEM STATEMENT

2.1 Graph Neural Networks and Autoencoders

Graph Autoencoders. Graph Autoencoders (GAEs) are unsupervised learning frameworks, consisting of a graph encoder and a graph decoder. The graph encoder learns network embeddings by mapping nodes into a latent vector, while the graph decoder learns to reconstruct the data from the encoded ones. Graph Convolution can be used to encode the nodes to produce a low dimensional network embeddings [18]. It aggregates signals from neighboring nodes to learn embeddings for each node:

$$\bar{X} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}XW) \quad (1)$$

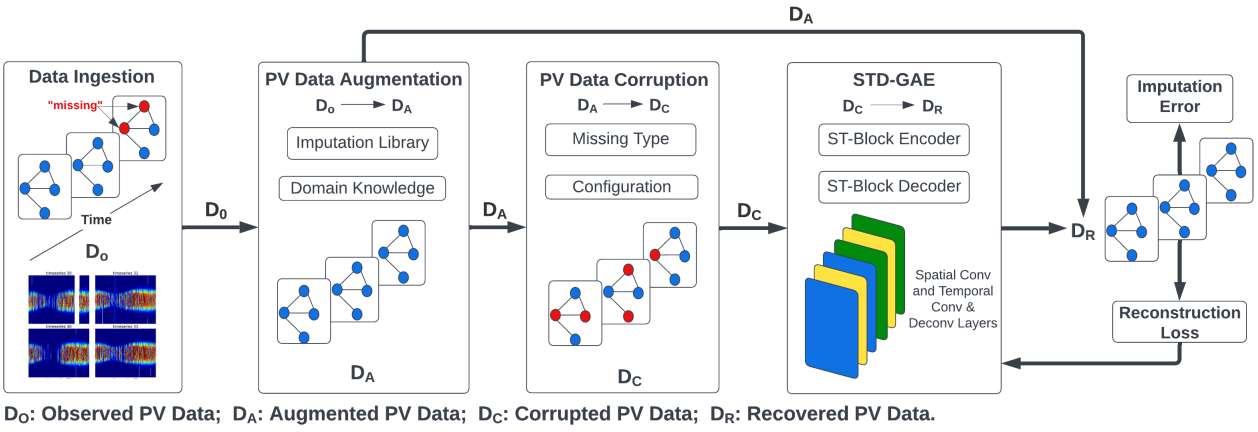


Figure 2: Overview of STD-GAE Imputation Framework.

where X is the node attribute matrix, $\tilde{A}$ is the adjacency matrix with self loop, $\tilde{D}$ is the degree matrix, W is a trainable weight matrix, and σ denotes the activation function.

Previous studies in GAEs have shown their outstanding performance in learning from corrupted data, which is the natural extension of the problem of missing data imputation [26]. STD-GAE integrates GAEs to learn the distribution of node and topological features to handle complex missing scenarios.

Denoising Autoencoders. Denoising Autoencoder (DAE) is a variant of Autoendoer (AE), with a goal to recover data from corrupted input. AE tends to overfit for data imputation, since it minimizes the reconstruction loss between the input and the reconstructed counterpart that may lead to an identity function. Instead, DAE introduces noise (corruption) to the input. Input data can be corrupted by noises added to the input vector in a stochastic manner [14]. The model is then trained to minimize the reconstruction loss between the recovered data and the uncorrupted counterpart.

2.2 PV Network Representation

PV Network. We represent the spatiotemporal PV data as an undirected graph $G = (V, E, X_t)$, where (1) each node in V represents a PV inverter; and (2) X_t denotes a node attribute tensor $\in \mathbb{R}^{T \times n \times d}$. Here T is the length of timeseries, n is the number of nodes (which is 98 in our study), and d is the number of input channel. Since the locations of PV inverters are fixed, the graph structure is static with time-invariant nodes and edges. However, X_t is time-varying: each node i carries a timeseries $x_i \in \mathbb{R}^{T \times d}$ recording attributes such as temperature, wind speed, irradiance and power output.

Modeling Edges. We represent edges by edge index as a tensor $E_{index} \in \mathbb{R}^{2 \times m}$ and edge weight as a tensor $E_W \in \mathbb{R}^m$. Here m is the number of edges. Both edge index and edge weight can be derived as follows:

$$W_{i,j} = \begin{cases} \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right), i \neq j \text{ and } \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) \leq \epsilon \\ 0 \text{ otherwise} \end{cases} \quad (2)$$

where d_{ij} is the Euclidean distance between the node pair (i, j). σ is the standard deviation of the distances. The network sparsity will be decided by ϵ. When $\epsilon = 0$, we will have all nodes connected with each other. As ϵ increases, the sparser the network will be.

PV data imputation. Given a PV network G with observed PV timeseries data $\{X_1, \ldots X_T\}$ as input, our goal is to develop a model to impute all the missing data in the input. We next introduce STD-GAE framework.

3 STD-GAE FRAMEWORK

We start with an overview of STD-GAE framework, and then present the details of its major modules: data augmentation, data corruption, spatio-temporal convolution and deconvolution.

3.1 Framework Overview

The STD-GAE framework, as illustrated in Fig. 2, consists the following four major components.

Data ingestion. STD-GAE first collects and transforms the raw input data from metering infrastructure of PV inverters to attributed data D_O. The PV data is stored in HBase supported by CRADLE, an HPC cluster at CWRU (see Section 5).

PV Data augmentation. It then performs a data augmentation module to transform D_O to augmented PV data D_A from D_O via a light-weighted imputation process. The goal is to provide complete and high quality data for training STD-GAE models. Specifically, (a) the imputation is cold-started by invoking a suite of basic imputation algorithms from a built-in Imputation library; (b) it then exploits domain knowledge, encoded as a set of value dependencies and rules, to validate the imputed values and recommend the most appropriate augmentation method.

PV Data corruption. A data corruption module is performed to inject missing data to D_A, by declaring specified missing patterns and configuration. The corrupted data D_C and the augmented data D_A serve as the input to the STD-GAE model.

Model training. In the model training phase, STD-GAE learns the denoising graph autoencoder model (simply referred to as STD-GAE) by minimizing reconstruction loss, i.e., the the mean squared

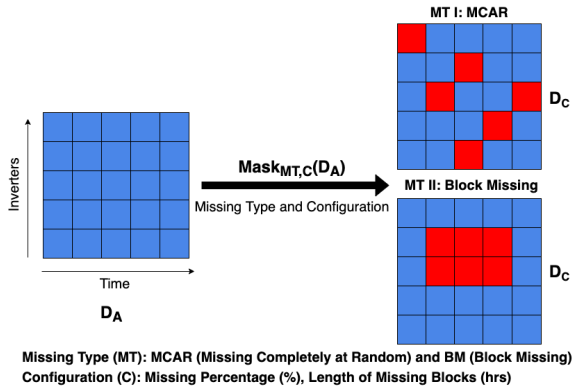
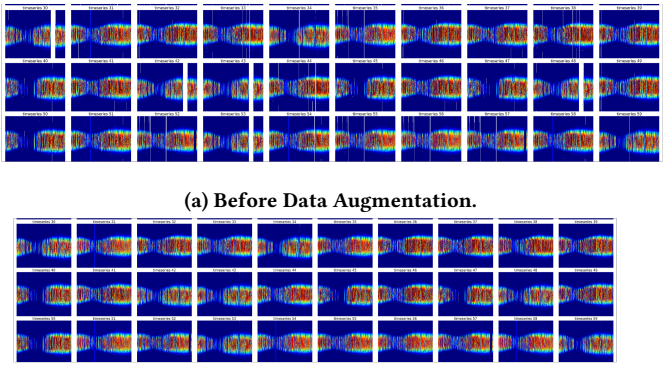


Figure 3: PV Data Corruption.



(a) Before Data Augmentation.

(b) After Data Augmentation.

Figure 4: Heatmaps of Daily Power Output of Randomly Sampled 30 PV Inverters.

error between the reconstructed data D_R (given the corrupted input D_C) and D_A. Let the set of training parameters be Θ, the value of recovered data for inverter i at time t as $D_R(t,i)$, the optimal parameter setting be θ, training loss be $L(D_A, D_C)$, and number of observations be $N = T \times n$, the learning objective is to minimizing the following loss function:

$$\theta = \arg\min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{T} \sum_{t=1}^{n} (D_R(t,i) - D_A(t,i))^2 \tag{3}$$

Note that the reconstruction loss is different from imputation error since the latter is only computed on the missing part of test data. Instead of focusing only on the reconstruction loss of missing data in the training stage, our trained model learn spatiotemporal correlations from the whole training data to better recover missing in the out-of-sample data. We train, validate, test our model in a sliding window fashion with both window size and step size set to be 288. Since each interval is five minitues, window size 288 amounts to a timeseries with a length of one day.

3.2 Data Augmentation and Corruption

Data Augmentation with Domain Knowledge. Since the denoising graph autoencoders in STD-GAE need to be trained on complete data, we utilize PV Data Augmentation to fill in the missing data to obtain D_A. STD-GAE has a built-in Imputation Library that contains a set of primitive imputation methods such as linear interpolation (LI), KNN, and MICE.

STD-GAE leverages domain Knowledge from PV scientists and engineers. For example, in PV industry, a simple Predicted Power model following the irradiance and temperature scaling approach at widely implemented [11].

$$P = \frac{G_{POA}}{1000} \frac{P_{norm}}{1 + \gamma_T (T_{module} - 25)} \tag{4}$$

where P is the estimated power, P_{norm} is the power of the PV module at standard testing condition, G_{POA} is the irradiance incident on the plane of the module or array (W/m^2), γ_T is the temperature coefficient of PV modules, and T_{module} is the module temperature ($^\circ C$). While this heuristic rule does not fully reflect all deterministic factors of T_{module} (*e.g.,* it may ignore the impact of low light, clipping loss, shading, and module degradation), STD-GAE leverages the value dependencies to validate impossible values. For example, the above equation is used to set a reasonable value ranges (domains) for the missing attribute to be imputed, along with other value constraints for *e.g.,* nameplate power of PV modules: $P_{nameplate} \geq P_{norm} \geq 80\% P_{nameplate}$; since a typical solar panel's performance warranty will guarantee 80% at 25 years. By default, STD-GAE chooses K-nearest Neighbor after validating the results with the domain knowledge rules. Fig. 4 illustrated PV power output values of 30 randomly sampled inverters before and after the validated data Augmentation.

Data Corruption. In the Data Corruption module, we generate missing masks to simulate the distribution of real-world missing patterns of PV data. Missing mask is characterized by a configuration tuple (MT, C), where MT denotes a set of desired missing types (*e.g.,* MCAR or BM), and C denotes a configuration parameters: missing rate, or length of block missing.

Fig. 3 shows an example of corruption phase using the above two missing patterns, where the red (resp. blue) blocks denote missing (resp. augmented) values. (1) For MCAR, missing data masks are generated using a uniform distribution between 0 and 1 such that the threshold corresponds to the selected configuration of missing percentage. All the missing values are randomly scattered which is typically caused by short-term power or communication failure. Thus, they are independent of each other. (2) For BM, missing data masks are created by injecting a fixed length to daily timeseries of each PV inverter. The fixed length is chosen according to the selected configuration of length of block missing. BM is more commonly caused by a longtime malfunction of the sensors.

According to the selected missing mask $Mask_{MT,C}$, we inject missing data points to D_A. If $Mask_{MT,C}(t,i) = 1$, then $D_C(t,i)$ is observed, while $D_C(t,i)$ is missing if $Mask_{MT,C}(t,i) = 0$. We represent corrupted data D_C as $D_C = Mask_{MT,C}(D_A) = D_A \odot Mask_{MT,C}$, where $\odot$ is the element-wise product operator.

3.3 Spatial-Temporal Blocks

We next detail our design of spatial layers and temporal layers in the ST blocks of STD-GAE.

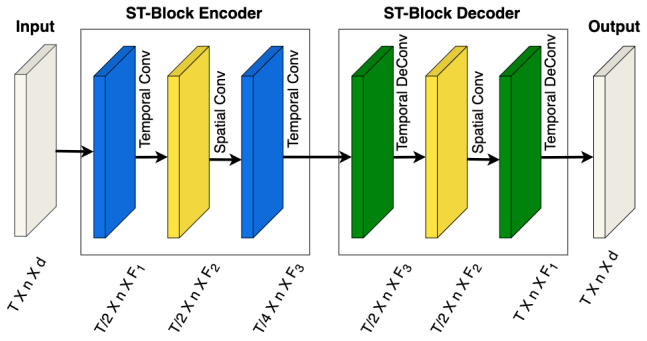


Figure 5: Structure of the Spatial Layers and Temporal Layers in the Proposed STD-GAE.

Spatiotemporal (ST) Block. Each ST Block consists of one spatial layer sandwiched between two temporal layers. Both decoder and encoder are composed of a ST Block. As illustrated in Fig. 5, the ST-Block encoder extracts the PV power characteristics by mapping input data into lower dimensional node embeddings, while the ST-Block decoder recovers original dimension of the reduced data.

Temporal Layers. To capture the temporal correlations, i.e., correlations in the timeseries of each PV inverter, we use gated 1D convolution in both encoder and decoder. A 1D filter $\Omega \in \mathbb{R}^K$ is used to perform convolutions on daily timeseries of each PV inverter by aggregating neighboring values in timeseries. $\Omega * x$ performs convolution on x such as its length is reduced to a half with $K = 4$, stride = 2 and padding = 1. Then a gated linear unit (GLU) is used as activation defined as:

$$GLU = (\Omega * x) \odot \sigma(\Omega * x) \tag{5}$$

where σ is a sigmoid function and $\odot$ is element wise product operator. After the temporal convolutional layers in encoder, deconvolutional layers in decoder restore length of timeseries to original size by performing reverse convolutions. Temporal convolutional and deconvolutional layers are designed in such a way that input and output of the STD-GAE have the same size.

Spatial Layers. A spatial convolutional layer performs convolution on a graph by aggregating data points from neighboring nodes in both encoder and decoder. We adopt the chebyshev spectral graph convolutional operator (ChebConv) [9] as our spatial convolutional layer. ChebConv is a spectral-based method. Let $\mathrm{L} = \mathrm{I_n} - \mathrm{D}^{-\frac{1}{2}}\mathrm{A}\mathrm{D}^{-\frac{1}{2}}$ be the normalized graph Laplacian matrix. Since L is real symmetric definite, it can factored as $\mathrm{L} = \mathrm{U}\Lambda\mathrm{U}^T$, where $\mathrm{U} \in \mathbb{R}^{n \times n}$ is the matrix of eigenvectors ordered by eigenvalues and Λ is the diagonal matrix of eigenvalues with $\Lambda_{ii} = \lambda_i$. The spectral convolution of a filter kernel $\mathbf{g}_\theta$ with signal X is defined as:

$$\mathrm{X} * G\mathbf{g}_\theta = \mathrm{U}\mathbf{g}_\theta\mathrm{U}^T\mathrm{X} \tag{6}$$

where $\mathrm{U}^T\mathrm{X}$ is a graph Fourier transform to signal X. Since the eigen-decomposition requires $O(n^3)$ computational complexity, ChebConv reduces the complexity to $O(m)$. ChebConv approximates $\mathbf{g}_\theta$ by Chebyshev polynomials of Λ, i.e., $\mathbf{g}_\theta = \sum_{i=0}^{k} \theta_i\, T_i(\tilde{\Lambda})$, where $\tilde{\lambda} = \frac{2\Lambda}{\lambda_{max}} - \mathrm{I_n}$. Since $T_i(\tilde{L}) = \mathrm{U}T_i(\tilde{\lambda})\mathrm{U}^T$, it can be shown that Equation 3 can be estimated by:

$$\mathrm{X} * G\mathbf{g}_\theta = \sum_{i=1}^{K} \theta_i T_i(\tilde{L})\mathrm{X} \tag{7}$$

where $\tilde{L} = \frac{2L}{\lambda_{max}} - \mathrm{I_n}$ and k is the size of the kernel for spatial convolution. The value K shows signals passed to nodes up to k hops away. In this study, we choose $k = 3$ to achieve a balance between cost and accuracy.

4 EXPERIMENTAL STUDY

4.1 Experimental Setup

Dataset. We use a dataset that contains normalized inverter alternating current power of 98 PV inverters from 8 PV sites in Canada, which covers 360 days ranging from 09/01/2015 to 08/25/2016. The dataset's sample interval is 5 minutes, amounting to 103,680 data points for each PV inverter timeseries. We split our dataset into training, validation, and testing as follows: day 1-240 for training, day 241-300 for validation, and day 301-360 for testing.

Evaluation Metrics. We use two classes of evaluation metrics.

Imputation accuracy. The imputation accuracy is evaluated by Mean Absolute Error (MAE) and Rooted Mean Squared Error (RMSE), and domain knowledge based metrics. For each tested missing scenario, a data corruption mask selected according to the selected missing data type (MCAR or BM) and configuration (missing rate or length of missing interval). We denote test data as D_T, the observed part of it as $D_{T_o} \subseteq D_T$, and the augmented test data as $D_{T'}$. The data corruption (mask) is only applied to a subset of $D_{T'}$ such that all the elements have counterparts in D_{T_o}. Denote $D_{T'_c} \subseteq D_{T'}$ as the corrupted data, imputed values of $D_{T'_c}$ as P, and the ground truth of $D_{T'_c}$ as $\tilde{P}$. Each metric can be defined as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|P_i - \tilde{P}_i|; \quad RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(P_i - \tilde{P}_i)^2} \tag{8}$$

where $N = card(D_{T'_c})$, $P_i \in P$, and $\tilde{P}_i \in \tilde{P}$.

Domain-specific Metrics. We also consider *percentage of outliers* and *seasonality*, two established domain-specific metrics for quantifying the quality of imputed PV data. [13, 16, 22, 27]. Specifically, we evaluate whether the imputation algorithms can "recover" the same level of outlier percentage and seasonality of the original dataset. (1) The outliers refer to the data points that are greater than ± 1.5 times the interquartile range. We adopt tsoutliers R package to detect outliers [23]. (2) It is natural to investigate whether the imputation methods are able to maintain seasonality, which is a common feature of PV timeseries. We use the STL function to decompose the each dataset into a seasonal component, a remainder component, and a trend using locally weighted nonparametric regression, and compare the seasonal component among datasets from various imputations. The STL functions are available in the base R stats package. Both outlier detection and seasonality characterization functions are available to our PVplr R package [1, 2].

Imputation Methods. We compare STD-GAE with 6 baselines.

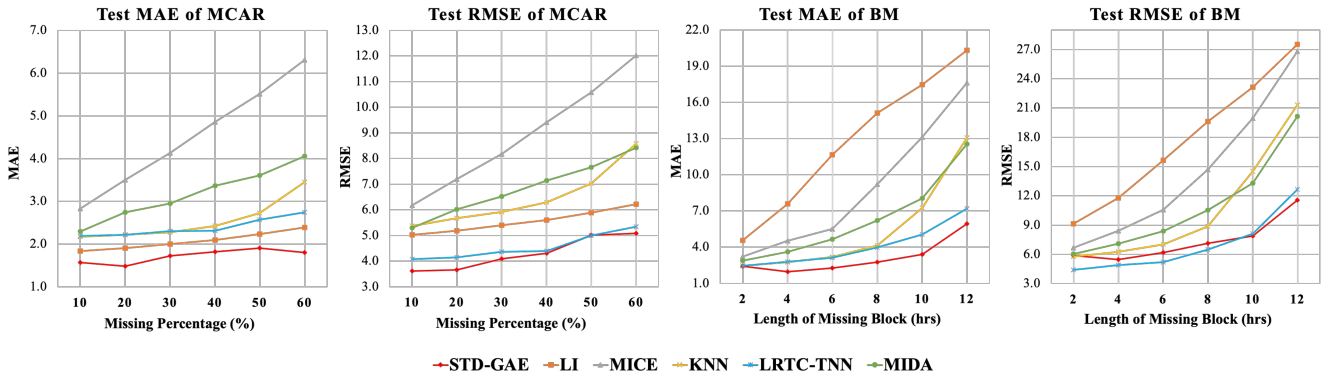


Figure 6: Imputation Errors and Impact of Missing Scenarios and Severity (results of Mean Imputation are out of scale).

(1) *Linear Interpolation (LI)* [5]: a timeseries imputation method that fits a simple linear model using two values before and after the missing data block. Each missing data point will then be estimated using the linear model.

(2) *Mean Imputation (Mean)* [12]: a common approach that uses the column-wise mean to fill the missing data.

(3) *KNN* [24]: imputes data by finding and averaging K nearest neighbors to fill in the missing value.

(4) *MICE* [29]: MICE makes multiple imputations using chained equations. It imputes missing values in the variables of a dataset by using a divide and conquer approach. Once the focus is placed on one variable, MICE uses all the other variables in the dataset to predict missing part of that variable. The prediction is based on a regression model, with the form of the model depending on the nature of the focus variable.

(5) *LRTC-TNN* [7]: Low-Rank Tensor Completion with Truncated Nuclear Norm minimization (LRTC-TNN) is one of the state-of-the-art spatiotemporal imputation technique designed initially for traffic data imputation. It formulates the data imputation problem in a low-rank tensor completion (LRTC) framework and defines a novel truncated nuclear norm (TNN) on spatiotemporal data in form of $location \times day \times time$. In particular, it introduces an universal rate parameter to control the degree to allow better characterize the hidden patterns in spatio-temporal data.

(6) *MIDA*: [14]: a denoising autoencoder approach that uses fully connected layers. Unlike the denoising graph autoencoders in our proposed STD-GAE, the encoder in MIDA maps the input data to a higher dimension while decoder maps back to original input. The model is trained on randomly corrupted inputs.

Configuration. The proposed STD-GAE is trained by Adam Optimizer [19]. The learning rate starts at 0.001 adjusted with a decay rate of 0.02. The ChebShev filter size is set to 3. The network sparsity parameter ϵ is set to 1. During the training stage, the batch size is set to 2 and the number of epochs is 50. Selected model is the one that minimizes validation loss. All experiments are implemented with Pytorch Geometric Temporal [30] and conducted on Intel Xeon(R) CPU E5-2630 v4 @ 2.20GHz, 64 GB Memeory, 6 CPU cores, and 12GB NVIDIA GeForce RTX 2080 GPU.

Our source code, and a full version of the paper is available[1].

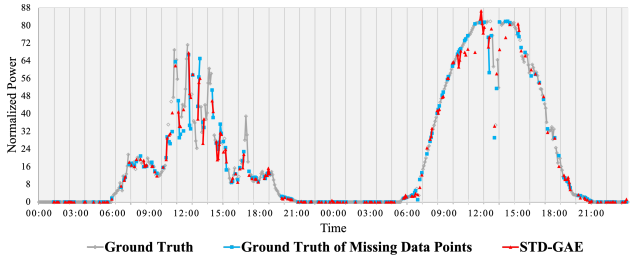
4.2 Experiment Results

Exp-1: Comparisons of Imputation Accuracy. We investigate two missing data types as mentioned in the Section 3.2, i.e., missing completely at random (MCAR) and block missing (BM). We conduct controlled experiments by varying the missing rate of MCAR from 10% to 60% and length of missing intervals of BM from 2 hours to 12 hours. In total, we test 12 different missing scenarioes. Fig. 6 reports MAE and RMSE of the methods.

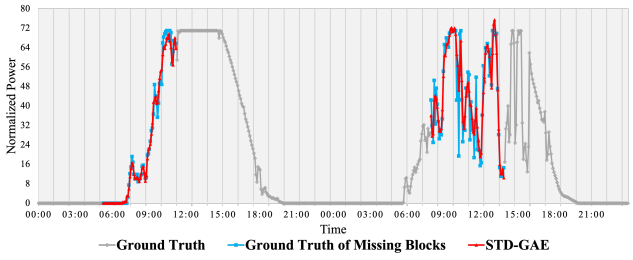
We have the following observations. (1) STD-GAE achieves the best imputation performance in most scenarioes. Second best LRTC-TNN achieves similar performance with STD-GAE only in the test RMSE of BM, but worse than STD-GAE in other cases. (2) All imputations perform better in MCAR than BM. Compared to MCAR, the increase of imputation error is the highest for LI and the lowest two are STD-GAE and LRTC-TNN. This finding indicates that STD-GAE can leverage spatial correlations of PV inverters to recover the large chunks of missing data when neighboring timestamps are not available. (3) For the same missing data type, as the severity (missing percentage or length of block missing) increases, STD-GAE maintains a comparable gain since it is trained by minimizing the reconstruction loss of whole training data to learn a better overall distribution of PV power data. Note that results of Mean Imputation are not shown here due to its errors being out of scale, i.e., delivering the worst performance in every missing scenario.

We randomly selected two PV inverters and visualized the imputation results, in two missing scenarios: 40% MCAR and 6-hours BM. Fig. 7 shows examples of imputed missing data of two PV inverters in two-days period compared to ground truth. In Fig. 7a where 40% random missing data are shown, STD-GAE can recover most of the missing observations and capture the majority of sharp fluctuations. In Fig. 7b where 6-hours block missing is presented, STD-GAE can accurately recover two 6-hours missing blocks by utilizing the spatial coherence from neighboring inverters.

[1]https://anonymous.4open.science/r/STD-GAE-B8FD

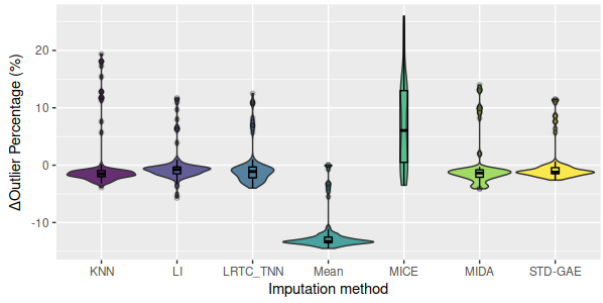


(a) Missing Completely at Random (MCAR).

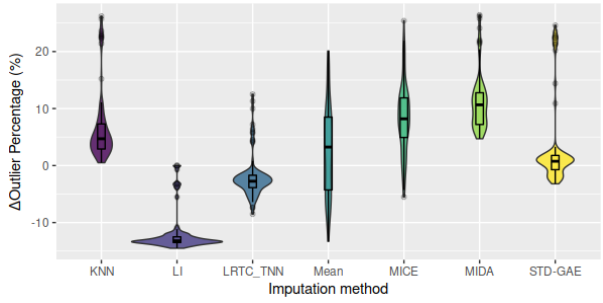


(b) Block Missing (BM).

Figure 7: Imputation Results of the Proposed STD-GAE.



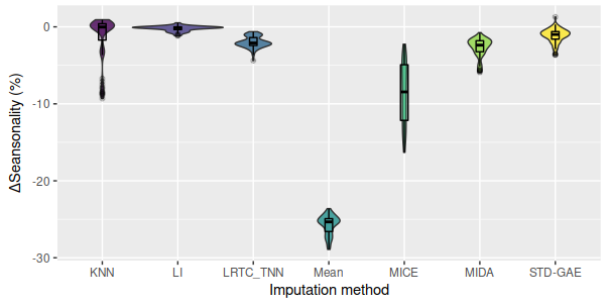
(a) Outliers Difference (60% MCAR).



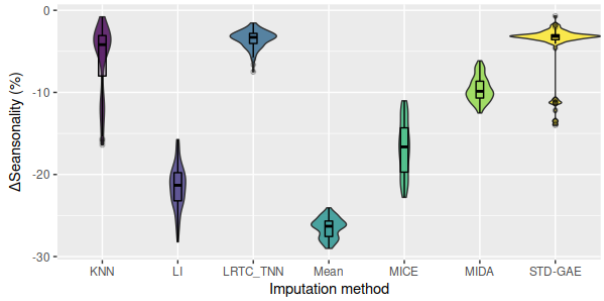
(b) Outliers Difference (12-hours BM).

Figure 8: Comparison of Outliers Recovery Using Different Imputation Methods for two Missing Types.

Exp-2: Comparison of Domain Knowledge-Based Metrics. To investigate whether the imputations could recover the outliers mainly induced by cloud shading, the difference between the recovered data using different imputations and the "ground truth" - pseudo imputed test data from Data Augmentation, for each individual inverter are presented in Fig. 8. Except for Mean and MICE,



(a) Seasonality Difference (60% MCAR).



(b) Seasonality Difference (12-hours BM).

Figure 9: Comparison of Seasonality Recovery Using Different Imputation Methods for two Missing Types.

all imputation methods are able to recover outlier percentage in MCAR. While only STD-GAE and LRTC-TNN perform well in BM, agreeing with the results in Fig. 6. However, all algorithms tend to generalize the results, introducing outliers to several inverters in locations that were far less often shaded by clouds.

Fig. 9 verifies the difference in seasonality feature between imputed data and test data. LI works well for MCAR but poorly for BM. For the two missing types, STD-GAE and LRTC-TNN outperform the rest. For 12-hours BM, LRTC-TNN outperforms STD-GAE at maintaining seasonality for places with large variance in power output between seasons, e.g., snow coverage reduced significant power loss in winter as compared to other seasons. On the other hand, STD-GAE utilizes seasonality features from neighboring inverters which are similar to each other, and performs best in most cases. In Fig. 9b, all data points are below 0 and none is able to recover exactly the same seasonality. This is due to a "worst case" where training and testing data are from different seasons (with quite different distributions). We are obtaining larger-scale PV datasets with length of more than 3 years to overcome this issue.

Exp-3: Case Analysis. We conduct ablation analyses to better understand the robustness of our proposed STD-GAE. First, we verify the robustness of our trained STD-GAE imputation model on different seasons. We select four out-of-sample monthly PV data for validation, including October 2016, January 2017, April 2017, and July 2017 as representatives from four seasons. We choose the missing type of 40% MCAR as the example for verification. The improvements in test RMSE compared to the closest baseline method are 27.79%, 23.77%, 19.27%, and 30.52% separately for these four months. Although the STD-GAE is not trained on a whole

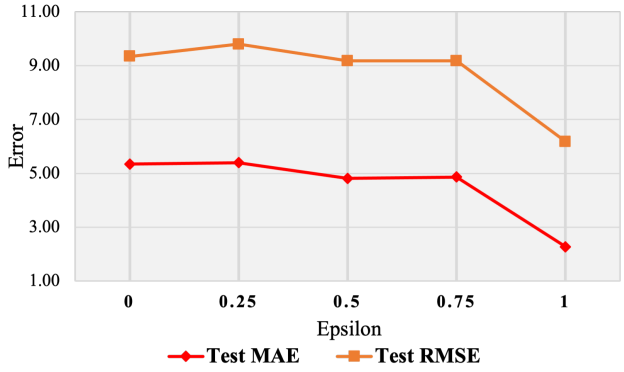


Figure 10: The Impact of Network Sparsity.

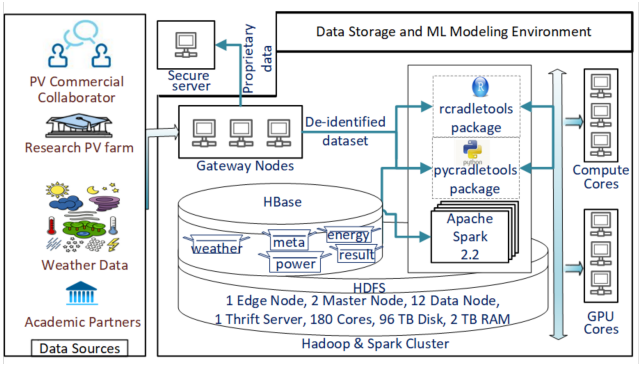


Figure 11: Data Workflow in CRADLE. A cron job periodically executes a program to collect PV data from different sources into Hadoop cluster. Pycradletools is the internal package developed to provide functionality to interact with HBase and data preprocessing in Python. Adapted from [15].

year's PV data, we still achieve relatively stable improvement in imputation accuracy for representative months from all seasons.

We also investigate the impact of network sparsity (ϵ) increases. We choose 6-hours BM as the example for comparison. As shown in Fig. 10, STD-GAE achieves the smallest errors when $\epsilon = 1$, *i.e.,* when all inverters at the same site are pairwisely connected (a clique) and there are no edges between inverters from different sites. This justifies our selection of edge models. On the other hand, the optimal ϵ may change upon the change of physical networks (*e.g.,* adding, moving or removing PV sites or inverters). STD-GAE can be readily updated accordingly for such changes.

5 FRAMEWORK DEPLOYMENT

The future application deployment of our proposed STD-GAE imputation model involves two steps below (Fig. 11).

PV Data Management. The PV data management consists of three steps: data acquisition, data preprocessing, and storage, similar to the previous work [17]. First, data comes from various commercial PV power plant companies. We collect these datasets through web APIs, secure shell FTP, or receiving them as CSV files over the cloud as encrypted zip files. We create cron jobs to run particular file parsers for datasets with different formats. Second, in data preprocessing, once data arrives, the first task is anonymization. We anonymize the proprietary information and save the anonymized data in Hadoop cluster. In the data processing step, timeseries from HDFS are read and passed through validation, tidying, and uniform structuring. Numerical values are checked for missing percentage or anomalies and assigned the quality score. Finally, only timeseries with high-quality score are ingested into HBase. Data in HBase are stored in a cell such that the value in a cell is uniquely identified by row, column qualifier, column family, and timestamp. To address the large overhead caused by millions of rows, the database is designed such that each cell contains a large amount of data compared to its unique identifier and keep one month of data as a string in cell.

Model Deployment. Our proposed STD-GAE can improve PV datasets for commerical plants, which will be integrated into the data preprocessing. Acquisition of real-time data at regular intervals is crucial for PV data imputation. The deployment of a production pipeline has to been integrated with the data management and inference modules. Data from HBase will be retrieved using pycradletools, a python package developed internally to provide an interface between the data acquisition and data preprocessing. All the imputation models will be trained on high performance clusters. The imputed data will be pushed to downstream inference module that includes the real-time PV power prediction model.

6 CONCLUSION AND FUTURE WORK

In this study, we have proposed Spatio-Temporal Denoising Graph Autoencoder (STD-GAE) that combines temporal correlations within-series and spatial correlations from neighboring PV inverters to recover missing data. Since STD-GAE is trained to minimize the reconstruction loss of the corrupted input, we can better learn spatiotemporal correlations and data distribution to recover missing data. We compared STD-GAE with existing methods in a real-world dataset from 98 PV inverters in Canada and obtain an improvement of 35.52% and 15.09% on average in the test MAE and test RMSE compared to the state-of-the-art missing data imputation methods like MIDA and LRTC-TNN. As the missing rate of MCAR or the missing block size of BM increases, the performance gap between the STD-GAE and baseline imputations become larger. Our tests also verified that STD-GAE retain data properties such as percentage of outliers and seasonality.

A future topic is to train STD-GAE on our newly ingested PV data in HBase that covers a span of at least three years. Another topic is to enable multiple channels for attributes temperature and irradiance in the PV production model, to further improve the performance of PV data imputation and PV degradation analysis.

ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under Solar Energy Technologies Office (SETO) Agreement Number DE-EE0009353. We thank Didier Thevenard from Canadian Solar Inc. for providing data.

REFERENCES

[1] Alan J. Curran, Tyler Burleyson, Sascha Lindig, David Moser, and Roger H. French,. 2020. PVplr: Performance Loss Rate Analysis Pipeline. https://CRAN.R-project.org/package=PVplr tex.ids: a.j.curranPVplrSDLEPerformance2020, curranPVplrPerformanceLoss2020.

[2] Alan J Curran, Tyler L Burleyson, Sascha Lindig, Joshua Stein, Laura S Bruckman, David Moser, and Roger H French. 2020. PVplr: R Package Implementation of Multiple Filters and Algorithms for Time-series Performance Loss Rate Analysis. In *PVSC 47*.

[3] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in Neural Information Processing Systems* 33 (2020), 17804–17815.

[4] Alessandro Betti, Maria Luisa Lo Trovato, Fabio Leonardi, Giuseppe Leotta, Fabrizio Ruffini, and Ciro Lanzetta. 2019. Predictive Maintenance in Photovoltaic Plants with a Big Data Approach. *ArXiv* (2019).

[5] Thierry Blu, Philippe Thévenaz, and Michael Unser. 2004. Linear interpolation revitalized. *IEEE Transactions on Image Processing* 13, 5 (2004), 710–719.

[6] Ajoy Kumar Chakraborty and Navonita Sharma. 2016. Advanced metering infrastructure: Technology and challenges. In *2016 IEEE/PES Transmission and Distribution Conference and Exposition (T D)*.

[7] Xinyu Chen, Jinming Yang, and Lijun Sun. 2020. A nonconvex low-rank tensor completion model for spatiotemporal traffic data imputation. *Transportation Research Part C: Emerging Technologies* 117 (2020), 102673.

[8] Jens Christiansen. 2021. *Global Market Outlook for Solar Power*. Technical Report. SolarPower Europe. 136 pages.

[9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*.

[10] Zulong Diao, Xin Wang, Dafang Zhang, Yingru Liu, Kun Xie, and Shaoyao He. 2019. Dynamic Spatial-Temporal Graph Convolutional Neural Networks for Traffic Forecasting. In *AAAI*.

[11] A. P. Dobos. 2014. *PVWatts Version 5 Manual*. Technical Report NREL/TP-6A20-62641. National Renewable Energy Lab. (NREL), Golden, CO (United States). https://doi.org/10.2172/1158421

[12] A Rogier T Donders, Geert JMG Van Der Heijden, Theo Stijnen, and Karel GM Moons. 2006. A gentle introduction to imputation of missing values. *Journal of clinical epidemiology* 59, 10 (2006), 1087–1091.

[13] Roger H. French, Laura S. Bruckman, David Moser, Sascha Lindig, Mike van Iseghem, Björn Müller, Joshua S. Stein, Mauricio Richter, Magnus Herz, Wilfried Van Sark, Franz Baumgartner, Julián Ascencio-Vásquez, Dario Bertani, Giosué Maugeri, Alan J. Curran, Kunal Rath, JiQi Liu, Arash Khalilnejad, Mohammed Meftah, Dirk Jordan, Chris Deline, Georgios Makrides, George Georghiou, Andreas Livera, Bennet Meyers, Gilles Plessis, Marios Theristis, and Wei Luo. 2001. *Assessment of Performance Loss Rate of PV Power Systems*. IEA-PVPS.

[14] Lovedeep Gondara and Ke Wang. 2018. Mida: Multiple imputation using denoising autoencoders. In *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 260–272.

[15] Ahmad Maroof Karimi, Yinghui Wu, Mehmet Koyuturk, and Roger H French. 2021. Spatiotemporal Graph Neural Network for Performance Prediction of Photovoltaic Power Systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[16] Arash Khalilnejad, Ahmad M. Karimi, Shreyas Kamath, Rojiar Haddadian, Roger H. French, and Alexis R. Abramson. 2020. Automated Pipeline Framework for Processing of Large-Scale Building Energy Time Series Data. *PLOS ONE* 15 (2020).

[17] Arash Khalilnejad, Ahmad M Karimi, Shreyas Kamath, Rojiar Haddadian, Roger H French, and Alexis R Abramson. 2020. Automated pipeline framework for processing of large-scale building energy time series data. *PloS one* (2020).

[18] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).

[19] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[20] Sascha Lindig, Atse Louwen, M Herz, J Ascencio-Vásquez, David Moser, and M Topic. 2021. Performance Imputation Techniques for Assessing Costs of Technical Failures in PV Systems. In *Proceedings / 38th European Photovoltaic Solar Energy Conference and Exhibition*.

[21] Sascha Lindig, Atse Louwen, David Moser, and Marko Topic. 2020. Outdoor PV system monitoring—input data quality, data imputation and filtering approaches. *Energies* (2020).

[22] Sascha Lindig, David Moser, Alan J. Curran, Kunal Rath, Arash Khalilnejad, Roger H. French, Magnus Herz, Björn Müller, George Makrides, George Georghiou, Andreas Livera, Mauricio Richter, Julián Ascencio-Vásquez, Mike van Iseghem, Mohammed Meftah, Dirk Jordan, Chris Deline, Wilfried van Sark, Joshua S. Stein, Marios Theristis, Bennet Meyers, Franz Baumgartner, and Wei Luo. 2021. International collaboration framework for the calculation of performance loss rates: Data quality, benchmarks, and trends (towards a uniform methodology). *Progress in Photovoltaics: Research and Applications* (2021).

[23] Javier López-de Lacalle. 2019. tsoutliers: Detection of Outliers in Time Series. https://CRAN.R-project.org/package=tsoutliers tex.ids: lopez-de-lacalleTsoutliersDetectionOutliers2016, lopez2016tsoutliers.

[24] R Malarvizhi and Antony Selvadoss Thanamani. 2012. K-nearest neighbor in missing data imputation. *International Journal of Engineering Research and Development* 5, 1 (2012), 5–7.

[25] Noor Bariah Mohamad, Boon-Han Lim, and An-Chow Lai. 2021. Imputation of Missing Values for Solar Irradiance Data under Different Weathers using Univariate Methods. *IOP Conference Series: Earth and Environmental Science* (2021).

[26] Ricardo Cardoso Pereira, Miriam Seoane Santos, Pedro Pereira Rodrigues, and Pedro Henriques Abreu. 2020. Reviewing autoencoders for missing data imputation: Technical trends, applications and outcomes. *Journal of Artificial Intelligence Research* 69 (2020), 1255–1285.

[27] Ethan M. Pickering, Mohammad A. Hossain, Roger H. French, and Alexis R. Abramson. 2018. Building electricity consumption: Data analytics of building operations with classical time series decomposition and case based subsetting. *Energy and Buildings* (2018).

[28] Irene Romero-Fiances, Andreas Livera, Marios Theristis, George Makrides, Joshua S. Stein, Gustavo Nofuentes, Juan de la Casa, and George E. Georghiou. 2022. Impact of duration and missing data on the long-term photovoltaic degradation rate estimation. *Renewable Energy* 181 (2022), 738–748.

[29] Patrick Royston and Ian R White. 2011. Multiple imputation by chained equations (MICE): implementation in Stata. *Journal of statistical software* 45 (2011), 1–20.

[30] Benedek Rozemberczki, Paul Scherer, Yixuan He, George Panagopoulos, Alexander Riedel, Maria Astefanoaei, Oliver Kiss, Ferenc Beres, Guzman Lopez, Nicolas Collignon, and Rik Sarkar. 2021. PyTorch Geometric Temporal: Spatiotemporal Signal Processing with Neural Machine Learning Models. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*.

[31] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. 2018. Structured sequence modeling with graph convolutional recurrent networks. In *International Conference on Neural Information Processing*. Springer, 362–373.

[32] Concepción Crespo Turrado, María del Carmen Meizoso López, Fernando Sánchez Lasheras, Benigno Antonio Rodríguez Gómez, José Luis Calvo Rollé, and Francisco Javier de Cos Juez. 2014. Missing data imputation of solar radiation data under different atmospheric conditions. *Sensors* (2014).

[33] Bohong Xiang, Feng Yan, Tao Wu, Weiwei Xia, Jin Hu, and Lianfeng Shen. 2020. An Improved Multiple Imputation Method Based on Chained Equations for Distributed Photovoltaic Systems. In *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*.

[34] Mao Yang, Dingze Liu, Yang Cui, Xin Huang, and Gangui Yan. 2020. Research on complementary algorithm of photovoltaic power missing data based on improved cloud model. *International Transactions on Electrical Energy Systems* 30, 7 (2020), e12350.

[35] Yongchao Ye, Shiyao Zhang, and James JQ Yu. 2021. Spatial-temporal traffic data imputation via graph attention convolutional network. In *International Conference on Artificial Neural Networks*. Springer, 241–252.

[36] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 3634–3640.

[37] Xiyue Zhang, Chao Huang, Yong Xu, and Lianghao Xia. 2020. Spatial-Temporal Convolutional Graph Attention Networks for Citywide Traffic Flow Forecasting. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*. 1853–1862.

[38] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. 2020. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 1234–1241.