Self-periodicity and environment based recurrent imputation for PV production data

Xiaolu Xu*, Muqing Ge†
* School of Computer Science, Minnan Normal University, Zhangzhou, China
† Department of Electrical and Electronic Engineering, the Hong Kong Polytechnic University, Hong Kong, China

***Abstract*—Missing measurement of PV production data is a common problem due to sensor breakdowns, server data corruption, and so on. Many imputation methods have been proposed, RNNs based methods are very popular among these methods due to the ability to capture temporal dependency, but they ignore periodicity features. Considering PV production data has a strong periodicity, we proposed an imputation method called SPERI, which combines Seasonal Trend decomposition using Loess (STL) with RNNs. Additionally, we adapt an encoder-decoder structure to introduce environment data. Specially, STL is used to compose periodicity features (such as daily and annual periods), and then the residual component is processed by RNNs. We evaluated our SPERI method on real-world datasets in point missing mode and consective missing mode, and SPERI achieves the highest scores in general.**

***Index Terms*—Deep learning, Data imputation, Seasonal trend loess, RNNs**

I. Introduction

Photovoltaic (PV) production data has an important role in predicting the area. However, missing value is a common problem during production and storage. This can be caused by a variety of reasons, such as sensor corruption, server data corruption, or inverter breakdowns.

Missing data will inevitably have a negative impact on reducing the quality of the data and making subsequent data analysis difficult. In order to improve the quality of the data, interpolation is necessary.

PV production data usually has a strong correlation with numerical weather data. In a real situation, photovoltaic power plants are usually equipped with a range of sensors to obtain Numerical Weather Data (NWD), such as irradiance, humidity, temperature, cloudiness, wind speed, and direction. NWD contains features that are difficult for the algorithm to predict, especially during sudden weather extremes. In these cases, the abrupt changes make it difficult for algorithms to fit the correct values[1]. The proposed method utilizes an encoder-decoder framework to effectively incorporate environment data from NWD. This approach enables our model to more accurately capture the dynamics of actual PV production data, thereby mitigating the risk of algorithmic misfit in the presence of sudden variations. Missing measurements in NWD are generally not a significant concern for the analysis of PV data, as NWD is typically rich and comprehensive. In cases where direct measurements are unavailable, numerical weather products from meteorological centers, such as the National Centers for Environmental Prediction (USA), the German Weather Service, and the Meteorological Service of Canada, can be utilized. These agencies provide robust numerical weather data, including data from neighboring stations, ensuring that relevant information can always be obtained by specifying the geographic location of the PV power plant. Therefore, this paper assumes that missing measurements only occur in PV production data.

In contrast to traditional interpolation of time series data, PV production data exhibit pronounced periodicity. PV production is strongly correlated with solar irradiance, which follows a distinct diurnal pattern—peaking at midday and approaching zero at night—as well as a cyclical annual pattern, with higher production in the summer and lower production in the winter. Additionally, seasonal variations influence the angle of solar incidence relative to the panel surface, further affecting the level of irradiance received by the panels. How to extract the periodicity features of PV production data is one of the main subjects of this paper.

Recently, with the popularity of deep learning, recurrent neural networks (RNNs) have been applied to time series data imputation as a way of modeling temporal dynamics[2]. RNNs can pass information from one moment to the next through a hidden layer, which gives RNNs an advantage when dealing with data with long-term dependencies. However, standard RNNs face the problems of gradient vanishing and gradient explosion when dealing with long-term temporal dependencies, which significantly limits their effectiveness in capturing remote dependency information[3]. To address this problem, improved RNNs variants such as Long Short-Term Memory Networks (LSTMs) and Gated Recurrent Units (GRUs) have been proposed, and these models effectively mitigate the gradient vanishing and exploding problems by introducing a gating mechanism, thus enhancing the ability to model long-time dependencies. Some articles [4] proposed that neural networks (NNs) are weak at modeling seasonality and trend which correspond to the periodicity feature we mentioned. Zhang et al.[5] proposed that the application of both detrending and deseasonalization can significantly improve the performance of neural NNs. In other words, NNs are better at dealing with stochastic features.

For the above consideration, we investigate different algorithms for time series data imputation with the objective of decomposing temporal features from the perspective of periodicity. For feature decomposition, Cleveland et al.[6] proposed seasonal-trend decomposition using Loess (STL), which can divide data into three components: seasonal, trend, and residual. The Fast Fourier Transform [7] also has the ability to capture periodicity features. RNNs are applied to capture stochastic features. This paper proposes a Self-Periodicity and Environment based Recurrent Imputation (SPERI) for PV production data, in which STL is used to capture temporal periodicity features to compensate for the insensitivity of RNNs to seasonality by decomposing data into seasonal, trend and residual components. Seasonal and trend components are the features of periodicity; the residual component will be trained in RNNs to reveal stochastic features. Additionally, the model has the structure of an encoder-decoder to learn the distribution of the environment, which is irradiance data. In our observation of PV data, we found that there is a strong positive correlation between PV power generation and irradiance, so in this paper, we claim that the distributional features of irradiance also apply to PV data. This method can establish features from both the environment and incomplete data, which allows the imputation to follow the existing distribution. We evaluate our method on a real-world dataset, which is a photovoltaic operation dataset[8]. It is suitable for scenarios where the severity of missing data is between 20% and 80%, while showing better performance than other baseline models. Experiment results show that our method achieves high imputation accuracy in terms of three evaluation metrics, including Root Mean Square Error (RMSE), R-Square, and mean absolute error (MAE). Among the methods above, our method achieves the best performance.

In summary, the contributions of our paper are as follows:

- We propose a new recurrent imputation method that accepts the periodicity feature and compensates for the insensitivity of NNs to seasonality. Specifically designed for PV production data, the method leverages an encoder-decoder framework to accurately model environmental distributions, demonstrating strong performance even with limited samples.
- An experiment is performed on two datasets that cover a missing rate between 20% and 80%, to evaluate the performance of our method in different missing scenarios. The results show our approach outperforms all baseline methods across various missing data scenarios.
- We release a new photovoltaic dataset from Qinghai, China, covering 365 days with a 15-minute sampling interval. This high-integrity dataset includes timestamps, real power, and irradiance, making it valuable for further research.

II. Related Work

Many imputation methods have been proposed to solve the missing value problem for time series data, particularly, there are articles dedicated to the application of algorithms in the field of photovoltaic. The simplest method is deleting missing data[9]. However, for continuous time series data, this method largely affects the analysis of the data, especially for fixed-step time series data. Statistical imputation methods fill the missing value with the mean value[10], the last observed value[11], and the mode value[12]. This method is easy to operate and reduces the impact of missing values on the dataset to some extent.

Abbreviations

SPERI	Self-Periodicity and Environment based Recurrent Imputation
PV	Photovoltaic
NWD	Numerical Weather Data
STL	Seasonal Trend Loess
RNNs	Recurrent Neural Networks
GRU	Gate Recurrent Unit
LSTM	Long Short-Term Memory
GAN	Generative Adversarial Network
ReLU	Rectified Linear Unit
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
KNN	K-Nearest Neighbor
MICE	Multivariate Imputation by Chained Equations
RF	Random Forest
ARIMA	AutoRegressive Integrated Moving Average
ARFIMA	AutoRegressive Fractionally Integrated Moving Average
SARIMA	Seasonal AutoRegressive Integrated Moving Average

For the past decade, some works show machine learning methods are useful for time series data imputation. The K-Nearest Neighbor (KNN) algorithm captures the k nearest neighbors for the mean value to fill in the missing measurement. Multivariate Imputation by Chained Equations (MICE) uses iterative regression equations to fill in missing values. Furthermore, Stekhoven et al.[13] proposed a method called miniForest based on Random Forest (RF), which shows better performance on their experiment than KNN in the missing range between 10% and 30%. For deep learning methods, recently, many researchers have focused on RNNs and generative adversarial network (GAN). RNNs reward population on time series data for their performance on short-term memory from neighboring messages. Autoreguressive methods[14] such as ARIMA[15], ARFIMA[16] and SARIMA[17] use self-relation between parameters to impute missing values. Panapakidis et al.[18] proposed a K-Means clustering method for imputing PV data. This method tries to find the nearest neighbor day curve of existing data for filling the curve of missing data.

Recently, deep learning has been considered the most promising approach to data imputation. Che et al. [19] proposed GRU-D, which uses training parameters to evaluate the time interval of the last observation for missing values. Luo et al.[20] proposed the GRUI (GRU for imputation), which has a simpler structure than GRU-D. Cao et al.[21] proposed BRITS (bidirectional recurrent imputation for time series),

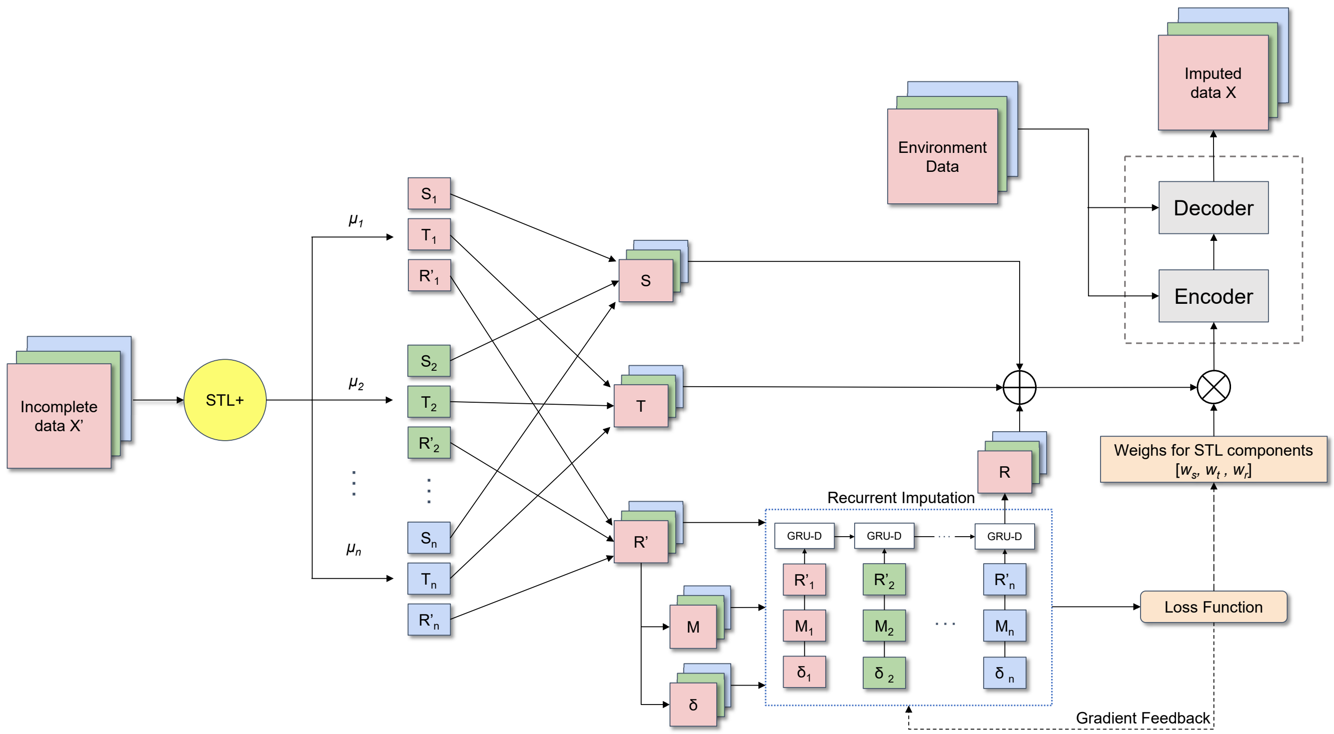

Fig. 1. A framework of SPERI

which adapts bi-directional structures. Luo et al.[22] proposed a GAN-based model called E^2GAN (End-to-End Generative Adversarial Network) to impute incomplete time series data by generating complete data from pretrained GAN.

III. Problem Statement

We denote a time series of length K as $X = (x_1, x_2, \dots, x_K)^{\mathrm{T}} \in \mathbb{R}^K$, where for each $k \in \{1, 2, \dots, K\}$,$x_k \in \mathbb{R}$ represents the k-th observed value. We introduce a masking vector $m_k \in \{0, 1\}$ to indicate the existence of X. The definition is as follows.

$$m_k = \begin{cases} 1, & \text{if } x_k \text{ is observed} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

To present the time lag from last observed value, we introduce a time interval vector $\delta_k \in \mathbb{R}$, and we define $s_k \in \mathbb{R}$ to present the timestamp. The definition is as follows.

$$\delta_k = \begin{cases} s_k - s_{k-1} + \delta_{k-1}, & \text{if } k > 1, m_{k-1} = 0 \\ s_k - s_{k-1}, & \text{if } k > 1, m_{k-1} = 1 \\ 0, & \text{if } k = 1 \end{cases} \tag{2}$$

In this paper, we focus on the problem of PV time series data imputation. In general, given the time series data with missing values as X', masking vector m and time interval vector δ as the input of model for training, finally we get a completed time series data X.

IV. Approach

In this section, we introduce our proposed method, Self-Periodicity and Environment-based Recurrent Imputation (SPERI). The foundation of our approach lies in addressing the known limitation of neural networks (NNs) in capturing periodic features, such as daily, weekly, and annual cycles. For instance, photovoltaic (PV) production data exhibits clear seasonal trends—decreasing during winter and increasing during summer—as well as diurnal patterns, with production peaking at noon and dropping to zero at night. Through our observation of PV datasets with missing values, we noted that at least 10% of the data typically remains intact, even in scenarios with significant data loss. This observation inspired our approach, wherein we use this 10% of complete data as the training set, allowing the model to learn intrinsic features from the data itself. The trained network is then used to impute the remaining 90% of the data. Our experiments demonstrated that this approach yields strong performance. Figure 1 shows the whole structure of SPERI.

A. Periodicity Decomposition

The framework of our method is shown in figure, consisting of three modules. The first module is used to separate features. To achieve this, we adopt STL, which is an efficient regression methods for feature extraction. According to the definition of STL[6], it decomposes time series data into seasonal, trend and residual components, indicated as s_k, t_k and r_k respectively for time series $X = (x_k, x_k, \dots, x_k)^{\mathrm{T}}, k \in \{1, 2, \dots, K\}$.

$$x_k = s_k + t_k + r_k \tag{3}$$

In STL, the length of seasonal is determined by parameter μ, after which STL can do local regression on the data slice in

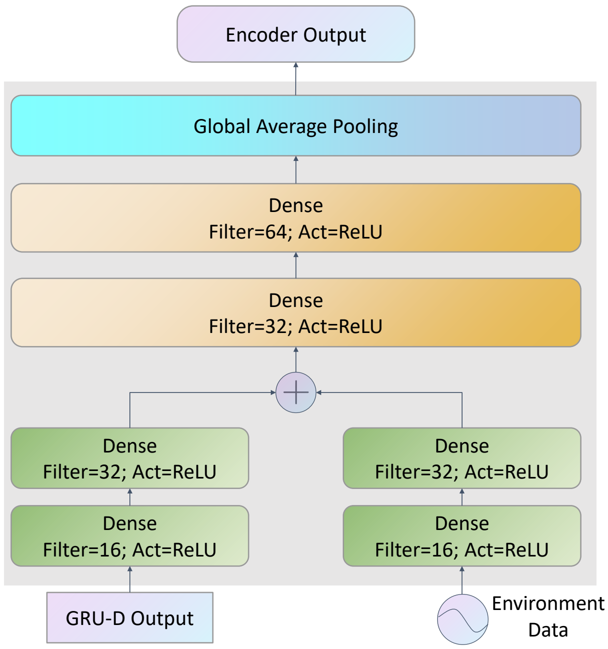

Fig. 2. The Encoder-Decoder structure

length of μ. By setting μ appropriately, we can ensure that the seasonal component accurately represents the recurring patterns within a day. Stated differently, the number of sample points in the data for a single day is the basis for the settling of the μ-value. The data is divided into a sub-sequence of cycles of length μ, each of which is a collection of data at a corresponding point in time during the day.For example, the dataset in our experiment has a 15-minute sampling interval which has 96 sample points during a day so $\mu = 96$.

We employ an enhanced STL method called STL+[23] to meet the requirements of the periodicity decomposition because the original STL algorithm lacks the mechanism to handle data with missing points. STL+ decomposes incomplete data into complete seasonal components S and T, and a incomplete residual component R'.

B. Recurrent Imputation

The goal of the Recurrent Imputation module is to deal with stochastic features. In detail, it imputes the missing values from R' component. We adapt GRU-D[19] as the core of the module in order to accomplish the goal. It is a cutting-edge recurrent imputation model that is based on the Gated Recurrent Unit (GRU).

Different from traditional GRU, GRU-D introduces a decay mechanism for input variables and hidden states by considering the distance from the last observed value. The decay rate γ is trainable, which is presented as

$$\gamma_k = \exp\{-\max(0, W_\gamma \delta_k + b_\gamma)\} \tag{4}$$

where W_γ and $b_\gamma)$ are trainable parameters of model. GRU-D settle decay parameters on measurement values by

$$r_k = m_k r'_k + (1 - m_k)(\gamma_k r'_{k'} + (1 - \gamma_{r'_k})\bar{r'}) \tag{5}$$

where $r_{k'}$ is the last observed value and $\bar{r}$ is the mean value. As we use GRU-D to process the residual component, we present the observation as r.

The last layer of the recurrent imputation module is a fully connected layer. The weights of the three components $W_f = \{w_s, w_t, w_r\}$ will update the loss function. The results of this module are as follows:

$$X = W(S + T + R) \tag{6}$$

where S and T are respectively the decomposed seasonal and trend components, R is the estimation of missing values by recurrent imputation module.

C. Encoder-Decoder Structure

The proposed method adopts an encoder-decoder fully connected system. The aim of this structure is to embed the values of the environment. In this structure, the input is a complete time series from the previous recurrent imputation and corresponding environment data. The reason is that we would like our model to have the ability to pick up features from environmental data. As stated in the introduction, there is a strong correlation between PV data and environmental factors, particularly irradiance. The environmental data includes extreme values caused by exceptional weather conditions, such as sudden drops in irradiance due to heavy rainfall, as well as consistently low irradiance levels throughout the day resulting from cloudy weather. This information is utilized to adjust and refine the output of the Recurrent Imputation process. In the following section, we examine the impact of this encoder-decoder structure on the overall performance and results.

Figure 2 shows the encoder-decoder structure. The inputs of the encoder are GRU-D results and environment data, through fully connected layers to expand dimensions, then averaged on the daily space axis.

The output of the encoder enters the decoder with the environment data. The environment data expends to high dimensions, the same as encoder output. Then the encoder output fits the number of daily space axis and takes an add operation. The result of the summation is passed to the dimension-reducing operation, and generates the imputed time series X. Every dense layer take Rectified Linear Unit (ReLU) function to transmit features. The dimensions of every input and output variables are shown in Table I and II. The value of K is equal to the number of observation points, which is 96 in our experiments.

TABLE I
DIMENSION OF VARIABLES OF ENCODER

	Variable	Dimension
Input	GRU-D Output	(BatchSize, K, 1)
	Environment Data	(BatchSize, K, n)
Output	Encoder Output	(BatchSize, 1, 64)

TABLE II
DIMENSION OF VARIABLES OF DECODER

	Variable	Dimension
Input	Encoder Output	(BatchSize, 1, 64)
	Environment Data	(BatchSize, K, n)
Output	Imputed time series X	(BatchSize, K, 1)

V. IMPUTATION METHOD

In this section, we compare our SPERI method with a set of baseline methods and a set of imputation methods based on decomposed methods. First, we take three baseline methods:

- **Zero**: The missing values are replaced with zeros.
- **KNN**: The missing values are imputed by the weighted average of K nearest neighbors.
- **ARIMA**[15]: The missing values are imputed by the moving average of integrated historical values.

To show the efficiency of our proposed SPERI method, we decomposed the function of our method, we obtained three imputed methods shown as follows:

- **GRU-D**[19]: An imputation method based on GRU. It introduces a decay mechanism to impute missing values by considering the distance from the last observed value.
- **STL-GRUD**: The time series data are first decomposed by STL to extract periodicity features, the the left residual component is processed by GRUD to capture stochastic features.
- **GRUD-Coder**: The combination of GRU-D algorithm and Encoder-Decoder structure. Environment data are considered compared with the original GRU-D.

VI. EXPERIMENT

In this section, we designed two experiments for the dataset. We considered two types of missing environments: point missing and conective missing. We compare our method with baseline methods and relative imputation methods. We conducted experiments with varying missing rates ranging from 20% to 80% to evaluate the performance of our model compared to baseline models. In our experimental setup, the first 10% of the data, which is complete and free of missing values, was designated as the training and validation set, while the remaining data, containing missing values, was used as the test set. In real-world scenarios, it is common to find at least 10% of the data that is complete. This allows for the extraction of these complete data segments and the subsequent reorganization of the dataset to better suit the requirements of our model. We adopt three evaluation metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and coefficient of determination (R2) to evaluate the performance of methods.

A. Data Standardization

The raw data are normalized by a linear variation. Respectively, for all elements in a time series, we have $\forall x \in [0, 1]$.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{7}$$

B. Point Missing

This experiment is mainly to evaluate the performance of our method and baseline methods in the setting of point missing situation. Data is dropped out at random points at a missing rate of $p\%$ ($p = 20, 30, \ldots, 80$).

Table III shows the performance of all imputing methods under point missing mode. Our SPERI method consistently outperforms across all missing rates, achieving the highest scores in every scenario. While the pure GRUD method performs well at low missing rates, its effectiveness significantly diminishes as the missing rate increases. Similarly, the STL-GRUD method exhibits a gradual decline in performance with higher missing rates. In contrast, both the GRUD-Coder and SPERI methods maintain strong performance across all missing rates, underscoring the critical importance of incorporating environmental data in scenarios with high levels of data loss.

C. Consective Missing

This experiment is mainly to evaluate the performance of our method and baseline methods in the setting of consective missing situation. In this situation, we generate consecutive missing values of 4 days (which means $4 * 96$ values in our dataset) as a large vacancy. We stipulate that for every 10% increase in the missing rate, there will be one more large

TABLE III

THE PERFORMANCE FOR ALL IMPUTATION METHODS UNDER POINT MISSING MODE

Miss Rate	Metric	Zero	KNN	ARIMA	GRUD	STL-GRUD	GRUD-Coder	SPERI
20%	RMSE	0.14084	0.11726	0.34771	0.11085	0.10570	0.08430	0.07972
	MAE	0.03488	0.04287	0.24252	0.06572	0.06053	0.04205	0.03954
	R2	0.71299	0.80102	0.75845	0.82069	0.83694	0.89629	0.90726
30%	RMSE	0.17432	0.14511	0.34076	0.10452	0.12425	0.09161	0.08198
	MAE	0.05291	0.06466	0.24140	0.05435	0.07394	0.04661	0.04099
	R2	0.56027	0.69529	0.65434	0.84057	0.77472	0.87752	0.90192
40%	RMSE	0.19992	0.16635	0.33041	0.11224	0.09746	0.09662	0.08379
	MAE	0.07013	0.08582	0.23750	0.06421	0.05499	0.04950	0.04203
	R2	0.42165	0.59956	0.57795	0.81617	0.86138	0.86377	0.89755
50%	RMSE	0.22443	0.18675	0.32030	0.14003	0.12228	0.08701	0.08134
	MAE	0.08802	0.10752	0.23296	0.08808	0.07199	0.04396	0.04072
	R2	0.27117	0.49536	0.47100	0.71384	0.78180	0.88950	0.90343
60%	RMSE	0.24385	0.20295	0.30667	0.15678	0.11258	0.09266	0.08347
	MAE	0.10471	0.12867	0.22805	0.09994	0.06403	0.04774	0.04131
	R2	0.13955	0.40399	0.37000	0.64127	0.81502	0.87471	0.89833
70%	RMSE	0.26358	0.21958	0.29485	0.14540	0.09470	0.09391	0.08085
	MAE	0.12200	0.15057	0.22487	0.08742	0.05280	0.04718	0.03992
	R2	0.00530	0.30231	0.26228	0.69146	0.86913	0.87129	0.90460
80%	RMSE	0.28286	0.23544	0.28332	0.15075	0.15545	0.09966	0.08797
	MAE	0.14023	0.17175	0.22109	0.09048	0.09527	0.05129	0.04406
	R2	-0.15780	0.19786	0.15843	0.66836	0.64733	0.85505	0.88707

TABLE IV

THE PERFORMANCE FOR ALL IMPUTATION METHODS UNDER CONSECTIVE MISSING MODE

Miss Rate	Metric	Zero	KNN	ARIMA	GRUD	STL-GRUD	GRUD-Coder	SPERI
20%	RMSE	0.14027	0.11658	0.35457	0.09651	0.14720	0.09521	0.08407
	MAE	0.03469	0.04229	0.24908	0.05132	0.08352	0.04590	0.04092
	R2	0.71527	0.80333	0.82955	0.86407	0.68379	0.86771	0.89687
30%	RMSE	0.17123	0.14276	0.33865	0.10717	0.10908	0.09255	0.08696
	MAE	0.05139	0.06333	0.23980	0.05671	0.06319	0.04661	0.04268
	R2	0.57575	0.70509	0.66258	0.83239	0.82634	0.87499	0.88963
40%	RMSE	0.19998	0.16606	0.33400	0.11805	0.10703	0.09726	0.08064
	MAE	0.06979	0.08452	0.24033	0.06907	0.06094	0.04888	0.03983
	R2	0.42132	0.60097	0.58644	0.79661	0.83282	0.86196	0.90511
50%	RMSE	0.22018	0.18337	0.31684	0.29509	0.10559	0.09976	0.08345
	MAE	0.08504	0.10461	0.23032	0.26322	0.05886	0.04797	0.04172
	R2	0.29846	0.51346	0.45320	-0.27083	0.83729	0.85476	0.89838
60%	RMSE	0.24180	0.20115	0.30788	0.11466	0.10177	0.09828	0.08693
	MAE	0.10230	0.12484	0.22792	0.06183	0.05667	0.04788	0.04364
	R2	0.15394	0.41449	0.36141	0.80813	0.84885	0.85905	0.88973
70%	RMSE	0.26006	0.21628	0.29745	0.14026	0.11338	0.10236	0.09547
	MAE	0.11864	0.14461	0.22477	0.08216	0.06382	0.04961	0.04737
	R2	0.02136	0.32314	0.27636	0.71292	0.81241	0.84710	0.86699
80%	RMSE	0.27734	0.23082	0.28670	0.29932	0.11956	0.11172	0.09446
	MAE	0.13463	0.16435	0.22174	0.27229	0.06843	0.05460	0.04920
	R2	-0.11304	0.22907	0.18256	-0.30749	0.79138	0.81785	0.86977

vacancy (with an initial value of one in 20% missing rate). Then we set a random drop to control the total missing rate at $p\%$ ($p = 20, 30, \ldots, 80$).

Table IV shows the performance of all imputation methods under consective missing mode. Our SPERI method achieves the highest score across all miss rates. Compared with the point miss mode, all methods have lower performance, especially in high miss rate. From this perspective, the STL-GRUD and SPERI methods demonstrate greater resilience to the consecutive missing data scenario, highlighting the importance of incorporating periodic features. The STL-GRUD method, by introducing periodicity into the model, shows a significant improvement over the pure GRUD, which suffers from mode collapse at certain missing rates. This collapse likely results from the combination of large data gaps and sporadic point drops, leading to unexpected vacancies that complicate the model's ability to leverage previous observations. The integration of periodic features in STL-GRUD helps mitigate this weakness. Additionally, the GRUD-Coder, which incorporates environmental data, maintains stable performance under these conditions. Our proposed SPERI method further enhances this approach by combining the advantages of both periodicity and

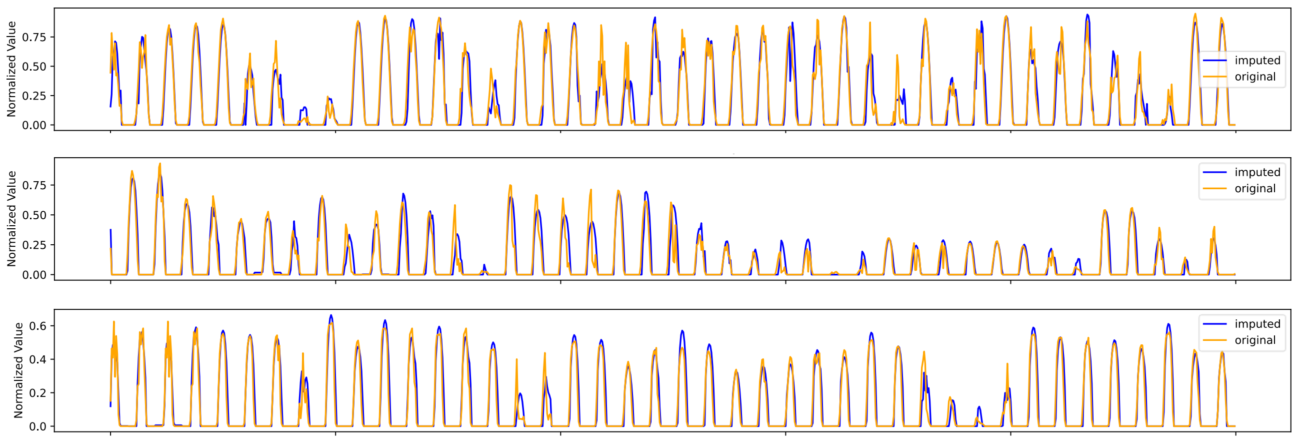

Fig. 3. Comparison of imputed data and original data (both are normalized)

environmental data integration, resulting in robust performance across varying missing data patterns.

VII. Discussion

Figure 3 shows the comparison of imputed data and original data. The plotted imputed data is point imputation with a 20% missing rate. The data was normalized, with each wave indicating one day's data. For the night hours, the imputed values are approaching zero. For the daytime hours, the imputed data appears to be smooth. This makes the output fit the slope, which takes up most of the data, making the output overall consistent with PV production. Production data usually has a distortion point at the peak, as shown in the figure, due to powerful sunlight irradiance at midday. Our method is able to learn this aberration feature to a certain extent, but there is a misfit for the simulation of the peaks. We considered that the possible reason is that the local peaks have a limited effect on the global, which makes our model ignore this in learning. Another possible reason is that the model is based on an encoder-decoder structure, which only captures salient features. However, the errors resulting from inaccurate peak forecasts are minimal, comprising only a small portion of the overall PV production data. On a broader scale, our method demonstrates a closer alignment with the true values, indicating its overall effectiveness in modeling the data. This demonstrates the ability of our model to capture global patterns.

VIII. Conclusion

In this paper, we have examined various time series imputation methods with a focus on effectively capturing the periodicity inherent in PV production data. We introduced SPERI, a novel imputation method that integrates Seasonal-Trend decomposition using Loess (STL) to extract periodic features, Recurrent Neural Networks (RNNs) to model stochastic components, and an encoder-decoder structure to incorporate environmental variables, thereby enhancing the overall accuracy and robustness of the model. Overall, our work has the following contributions:

- SPERI uses STL to extract periodicity features from data and leaves residual components for RNNs to extract stochastic features. Additionally, we adopt an encoder-decoder structure to introduce environment data.
- SPERI provides a mechanism for PV data to learn the self-periodicity, which includes daily period and annually period.
- We set up experiments with real-world datasets for our method and baseline methods. The result shows that SPERI achieves higher scores under point and consective modes than any other baseline methods.
- We publicize a real-world dataset for the benefit of other researchers.

For future work, we expect to introduce a self-supervised structure like GAN[24] to overcome the dependability of the availability of complete data. This may mean that we have to focus on the loss equation.

Data Availability

Data support for this paper comes from the Shandong University Power System Economic Operation Team[25]. We declared we had not changed the data. Our data set and the code for the proposed method are publicly available at https://github.com/Sirius-learning/SPERI.

References

[1] I. de-Paz-Centeno, M. T. García-Ordás, Ó. García-Olalla, and H. Alaiz-Moretón, "Imputation of missing measurements in PV production data within constrained environments," *Expert Syst. Appl.*, vol. 217, p. 119510, May 2023.

[2] Y. Zhou, J. Jiang, S.-H. Yang, L. He, and Y. Ding, "MuSDRI: Multi-Seasonal Decomposition Based Recurrent Imputation for Time Series," *IEEE Sens. J.*, vol. 21, no. 20, pp. 23 213–23 223, Oct. 2021.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[4] H. Hewamalage, C. Bergmeir, and K. Bandara, "Recurrent Neural Networks for Time Series Forecasting: Current status and future directions," *Int. J. Forecast.*, vol. 37, no. 1, pp. 388–427, Jan. 2021.

[5] G. P. Zhang and M. Qi, "Neural network forecasting for seasonal and trend time series," *Eur. J. Oper. Res.*, vol. 160, no. 2, pp. 501–514, 2005.

[6] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "Stl: A seasonal-trend decomposition," *J. Off. Stat*, vol. 6, no. 1, pp. 3–73, 1990.

[7] H. J. Nussbaumer and H. J. Nussbaumer, *The fast Fourier transform*. Springer, 1982.

[8] S. U. P. S. E. O. Team, "Wind power and photovoltaic operation dataset," https://energymeteo-pseo.sdu.edu.cn/dataset/, 2023, accessed on January 26, 2024.

[9] J. W. Graham and J. W. Graham, "Missing data analysis: making it work in the real world." *Annu. Rev. Psychol.*, 2009.

[10] M. Kantardzic, *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons, 2011.

[11] M. Amiri, M. Amiri, M. Amiri, R. Jensen, and R. Jensen, "Missing data imputation using fuzzy-rough methods," *Neurocomputing*, 2016.

[12] A. Purwar, A. Purwar, S. K. Singh, and S. K. Singh, "Hybrid prediction model with missing value imputation for medical data," *Expert Syst. Appl.*, 2015.

[13] D. J. Stekhoven and P. Bühlmann, "Missforest—non-parametric missing value imputation for mixed-type data," *Bioinformatics*, vol. 28, no. 1, pp. 112–118, 2012.

[14] W. Galbraith, John and V. Zinde-Walsh, "Autoregression-based estimators for ARFIMA models," 2001.

[15] D. Piccolo, "A distance measure for classifying arima models," *J. Time Ser. Anal.*, vol. 11, no. 2, pp. 153–164, 1990.

[16] V. Reisen, B. Abraham, and S. Lopes, "Estimation of parameters in arfima processes: A simulation study," pp. 787–803, 2001.

[17] C. Hamzaçebi, "Improving artificial neural networks' performance in seasonal time series forecasting," *Inf. Sci.*, vol. 178, no. 23, pp. 4550–4559, 2008.

[18] I. P. Panapakidis, A. S. Bouhouras, and G. C. Christoforidis, "A missing data treatment method for photovoltaic installations," in *2018 IEEE International Energy Conference (ENERGYCON)*. IEEE, 2018, pp. 1–6.

[19] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent Neural Networks for Multivariate Time Series with Missing Values," *Sci. Rep.*, vol. 8, no. 1, p. 6085, Apr. 2018.

[20] Y. Luo, X. Cai, Y. Zhang, and J. Xu, "Multivariate time series imputation with generative adversarial networks," vol. 31, 2018.

[21] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, "Brits: Bidirectional recurrent imputation for time series," *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.

[22] Y. Luo, Y. Zhang, X. Cai, and X. Yuan, "E^{2}GAN: End-to-End Generative Adversarial Network for Multivariate Time Series Imputation," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. Macao, China: International Joint Conferences on Artificial Intelligence Organization, Aug. 2019, pp. 3094–3100.

[23] R. P. Hafen, "Local regression models: Advancements, applications, and new methods," Ph.D. dissertation, 2023.

[24] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, 2018.

[25] S. U. P. S. E. O. Team, "Wind/photovoltaic operational data set," 2023. [Online]. Available: https://energymeteo-pseo.sdu.edu.cn/dataset