Renewable Energy and Power Quality Journal
https://repqj.com/index.php/repqj/index
RE&PQJ, Vol. 23, No. 1, 2025

Data Recovery of Distributed Power Station Output Considering the Output Information of Correlated Stations

Haiyan Zeng[1], Xiangli Peng[2], Chenxi Dong[2], Ran Wu[1], Jinqiang Lin[1,*], Yaqi Wang[1]

[1]State Grid Hubei Electric Power Co., Ltd. Xiangyang Power Supply Branch, Xiangyang, Hubei, China
[2]State Grid Hubei Electric Power Co., Ltd ., Wuhan, Hubei, China
[*]Corresponding author's email: mrbao5161@163.com

Abstract. The data acquisition devices of distributed photovoltaic power stations often lack proper maintenance, leading to frequent output data loss. This paper proposes a data recovery method that leverages the output information of correlated stations. First, the correlation between PV stations within the same region is calculated based on historical output data, and highly correlated station datasets are selected. Then, the complete data from these stations and the missing data from the target station are integrated and input into neural network models for recovery. Experimental results on the Desert Knowledge Solar Centre dataset show that incorporating correlated station data significantly improves accuracy. The TCN model achieves a 71.50% improvement, and the GRU model achieves 55.82%, outperforming other models due to their ability to capture temporal dependencies. This study's novelty lies in utilizing correlated station output instead of meteorological data, making it more practical for real-world PV data recovery.

Key words. Distributed Photovoltaic, Data Recovery, Neural Networks, Correlated Stations, Time-Series Imputation

Nomenclature

$\boldsymbol{X}$	Output data of a PV station
$\boldsymbol{Y}$	Output data of a PV station
$\boldsymbol{M}$	Mask matrix that indicates missing data positions
t	Time step index
i, j	Indices for PV stations i and j
r	Pearson correlation coefficient
$\boldsymbol{W}$	Weight matrix in the NN models
b	Bias term in the NN models
σ	Sigmoid activation function
h	Hidden state of the NN models
$\hat{y}$	Predicted value for missing data
$f(X)$	Function used to transform input data $\boldsymbol{X}$
r_{ij}	Correlation value between output data

1. Introduction

In recent years, distributed photovoltaic (PV) systems have seen rapid growth due to national policy support and technological advancements. However, compared to centralized PV power stations, distributed PV stations are primarily installed on the user side, typically with smaller capacities, fewer data acquisition devices, and insufficient maintenance. These limitations make them more prone to data loss, which can severely impact the accuracy of grid scheduling, output forecasting [1,2], and fault diagnosis tasks. Consequently, missing data presents a significant challenge for distributed PV systems.

Handling missing data in distributed PV systems can be approached in two main ways: direct deletion and imputation. Direct deletion is appropriate when the missing data constitutes a small proportion of the total dataset and does not disrupt the overall integrity of the data. However, for distributed PV output data, direct deletion disrupts the temporal periodicity of the data, and considering the limited data collection due to cost constraints, deletion further exacerbates data scarcity. This highlights the importance of imputation methods, which estimate missing values based on available data.

Imputation methods can be broadly classified into two categories: statistical-based methods and machine learning-based methods. Statistical-based methods, such as mean imputation, median imputation, linear interpolation, and spline interpolation, focus on the statistical properties of the data and generally require high smoothness. These methods, however, are unsuitable for recovering distributed PV output data due to the temporal fluctuations and variability in PV output that cannot be captured by simple statistical models [3]. In contrast, machine learning-based algorithms, such as K-nearest neighbors (KNN) [4], support vector machines (SVM) [5], random forests [6], and backpropagation (BP) neural networks [7,8], have gained widespread attention for their ability to capture complex patterns in data. For

example, bidirectional recurrent interpolation networks have been applied to model distributed PV clusters using grid-based models, incorporating meteorological data to achieve high-precision missing data recovery [9]. In [10], a modified Generative Adversarial Network (GAN), known as SolarGAN, was proposed for PV output recovery, outperforming traditional GAN models [11-16]. Additionally, numerical weather prediction (NWP) techniques have been used to restore missing PV data [17], while a time-multimodal variational autoencoder model integrated sky images, meteorological data, and PV output to enable recovery under varying missing data rates [18].

Despite these advances, practical challenges persist. Many distributed PV stations lack access to meteorological data acquisition systems or sky imaging technologies. As a result, data recovery methods are often forced to rely solely on historical output data. Satellite cloud images and numerical weather forecasts, when available, often fail to provide the necessary spatiotemporal resolution to effectively aid in data recovery. Furthermore, distributed PV stations are typically located closer together compared to centralized systems, which means that neighboring stations often experience similar environmental conditions. Given that PV output is highly correlated with meteorological factors, the output of neighboring PV stations also shows a strong correlation [19]. However, traditional methods for missing data imputation rarely consider the potential value of utilizing the output data from these neighboring stations.

This paper addresses this gap by exploring the role of correlation between adjacent distributed PV stations in improving the accuracy of missing data recovery. By calculating the historical output correlation between neighboring PV stations, the proposed approach selects a set of complete data from correlated stations that are highly related to the station with missing data. The output data from these correlated stations are then used to estimate the missing values, demonstrating a significant improvement in imputation accuracy.

The main contributions of this paper are as follows:

1. A novel data imputation strategy for distributed PV systems that incorporates the output data from correlated stations for missing data recovery.

2. A detailed analysis of how correlated station data can significantly enhance the accuracy of missing data recovery across multiple neural network models.

3. The validation of the proposed method using real-world PV datasets, confirming its practical effectiveness.

2. Correlation Analysis and Model Introduction

A. Distributed PV Correlation Analysis

The fundamental principle of PV power generation lies in the conversion of solar energy into electrical energy through the PV effect. As a result, the output of PV stations exhibits a strong positive correlation with solar irradiance, as illustrated in Figure 1. In addition to irradiance, the power output of PV modules is also influenced by various factors, including temperature, humidity, and wind speed. In general, the output of a PV station demonstrates a significant correlation with the prevailing meteorological conditions, underscoring the critical role of meteorological information in the analysis of PV data.

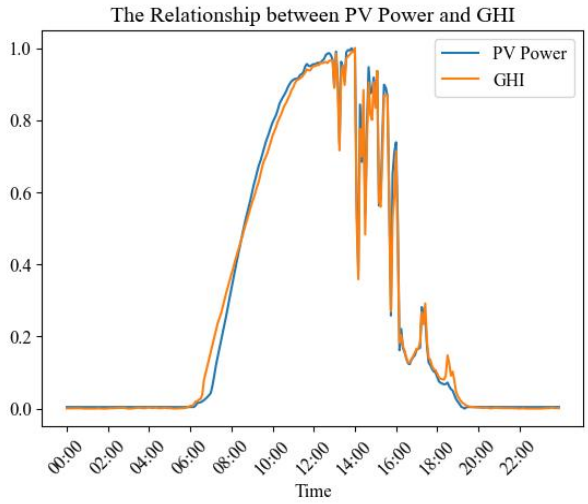

Figure 1. Correlation between Photovoltaic Output and Global Horizontal Irradiance (GHI)

However, most distributed PV systems are located on the user side and, due to cost constraints, generally lack meteorological data collection devices. As a result, only historical output data is available for analysis. As illustrated in Figure 2, adjacent PV stations, due to their similar environmental conditions, exhibit comparable patterns of output fluctuations. Therefore, when performing missing data imputation for distributed PV stations, the output data from neighboring, fully operational stations can provide valuable and reliable information to support the recovery of missing data at the target station.

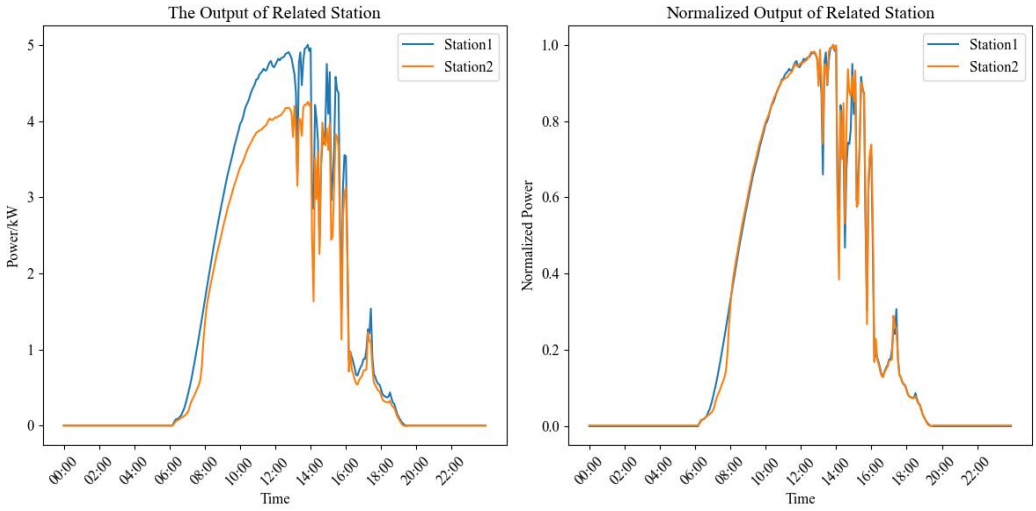

Figure 2. Output Curves of Adjacent Photovoltaic Stations

B. Model Introduction

To validate the role of neighboring stations in enhancing the accuracy of missing data imputation for distributed PV systems, this study conducts tests on several commonly used neural network models. The following sections provide a detailed introduction to these models.

1) Long Short-Term Memory Network

Long Short-Term Memory (LSTM) networks are a specialized type of Recurrent Neural Networks (RNNs) designed to address the vanishing and exploding gradient problems encountered by standard RNNs when processing long sequence data [20-22]. LSTM networks incorporate gating mechanisms that allow them to effectively learn and retain long-term dependencies in data.

The core idea of LSTM is to introduce memory cells and three gates (forget gate, input gate, and output gate) to regulate the flow of information, enabling it to pass through the network without being easily lost. The memory cell is the central component of LSTM, transmitting information across time steps. By selectively updating information, LSTM can retain and propagate useful memories at each time step, thus overcoming the gradient vanishing problem that traditional RNNs face when dealing with long sequences. The forget gate controls which information should be discarded from the memory cell, the input gate determines how much of the new information at the current time step should be stored, and the output gate controls how much the current memory state influences the output.

The operational process of LSTM can be divided into the following steps:

Forget Gate: Determines which information to discard. It takes the previous hidden state h_{t-1} and the current input x_t, and through the Sigmoid activation function, computes the proportion of information to forget, as shown below:

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \quad (1)$$

Where f_t represents the output of the forget gate, σ is the Sigmoid activation function, W_f denotes the weights, and b_f is the bias.

Input Gate: The input gate determines which new information should be stored in the memory cell, comprising two parts: deciding the values to be updated and generating the new candidate memory, as shown below:

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \quad (2)$$

$$\tilde{C}_t = \sigma\left(W_c \cdot \left[h_{t-1}, x_t\right] + b_c\right) \quad (3)$$

Where i_t represents the output of the input gate, indicating the proportion of new information to be stored in the memory cell at the current time step, and $\tilde{C}_t$ represents the candidate memory value, which is the new information that may potentially be added to the memory cell.

Updating the Memory Cell: The memory cell state is updated based on the outputs of the forget gate and the input gate, as follows:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (4)$$

Where C_t represents the current state of the memory cell, and C_{t-1} represents the state of the memory cell at the previous time step.

Output Gate: The output gate determines the amount of information to be output from the memory cell:

$$o_t = \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right] + b_o\right) \quad (5)$$

$$h_t = o_t \cdot \tan h\left(C_t\right) \quad (6)$$

Where o_t represents the output of the output gate, and h_t denotes the current hidden state, which will be passed to the next LSTM unit at the subsequent time step.

2) Gated Recurrent Unit

The Gated Recurrent Unit (GRU) is a variant of the Long Short-Term Memory (LSTM) network [23-26], which simplifies the LSTM structure by utilizing fewer gating mechanisms to enhance computational efficiency, while still retaining the ability to capture long-term dependencies. Unlike LSTM, GRU employs two primary gates: the update gate and the reset gate.

The update gate determines which parts of the current state should be inherited from the previous state and which parts should be updated by the current input information. Through the update gate, GRU is able to achieve long-term memory capabilities similar to those of LSTM, while avoiding the computational complexity associated with LSTM's multiple gates. The reset gate, on the other hand, determines the extent to which the current input influences the network's current state. It aids the network in "forgetting" irrelevant information, thus preventing the influence of outdated information on the current computation.

The computational process of the GRU is as follows:

Reset Gate: The reset gate determines how much of the old memory should be forgotten. It takes the previous hidden state h_{t-1} and the current input x_t , as shown below:

$$r_t = \sigma\left(W_r \cdot \left[h_{t-1}, x_t\right] + b_r\right) \quad (7)$$

Where r_t represents the output of the reset gate, σ denotes the Sigmoid activation function, W_r represents the weights, and b_r denotes the bias.

Update Gate: The update gate determines the extent to which the current state is influenced by the previous state and the current input:

$$z_t = \sigma\left(W_z \cdot \left[h_{t-1}, x_t\right] + b_z\right) \quad (8)$$

Where z_t represents the output of the update gate.

Candidate Activation: The candidate activation is controlled by the reset gate, determining the state of the candidate activation at the current time step:

$$\tilde{h}_t = \sigma\left(W_h \cdot \left[r_t \cdot h_{t-1}, x_t\right] + b_h\right) \quad (9)$$

Where $\tilde{h}_t$ represents the candidate activation.

Output: The updated state at the current time step is synthesized by weighting the previous time step's state and the current time step's candidate activation according to the update gate's weight:

$$h_t = \left(1 - z_t\right) \cdot h_{t-1} + z_t \cdot \tilde{h}_t \quad (10)$$

3) Temporal Convolutional Network

Temporal Convolutional Network (TCN) is a time-series data modeling method based on Convolutional Neural Networks (CNNs) [27-29], specifically designed for processing sequential data. Unlike traditional Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, TCN replaces the recurrent structure with convolutional layers, allowing for parallel processing of time-series data and mitigating the gradient vanishing or explosion issues that RNNs and LSTMs may encounter in long sequences.

The core idea of TCN is to model the dependencies within time-series data through causal convolutions and dilated convolutions. Causal convolution ensures that the output only depends on the current and past input data, preventing future information from leaking. Dilated convolution increases the receptive field by inserting gaps (i.e., dilation rate) into the convolutional kernel, enabling the model to capture longer-range dependencies with fewer convolutional layers. By increasing the dilation rate, convolution operations can cover a broader time-series range without the need to add more convolutional layers.

The basic building block of TCN, as illustrated in Figure 3, consists of dilated convolution layers, weight normalization layers, ReLU activation function layers, and Dropout layers. Residual connections are employed between the input and output of each basic unit, a design choice that helps alleviate gradient vanishing or explosion issues often encountered in deep networks.

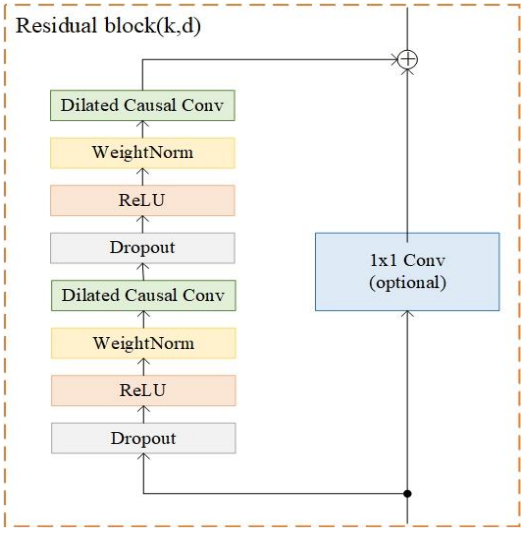

Figure 3. Basic Unit of TCN

4) Transformer

The Transformer model was introduced by Vaswani et al. in 2017 [30-33]. It is a model based on the self-attention mechanism, which discards the traditional Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs), enabling more efficient processing of sequential data.

The core idea of the Transformer is to use the self-attention mechanism to capture contextual information within sequence data without relying on recurrent or convolutional operations. Its structure can be divided into an encoder and a decoder, with each part consisting of multiple stacked sub-layers, as shown in Figure 4. These sub-layers mainly include the self-attention layer and the feed-forward neural network layer. Unlike RNNs, Transformers do not have sequential dependencies, allowing for parallel processing of all positions in the input sequence, greatly improving computational efficiency. Through the self-attention mechanism, the Transformer can effectively capture long-term dependencies, unlike RNNs, which are limited by sequence length. Since it does not rely on recurrent or convolutional operations, the Transformer offers greater flexibility in handling different types of sequential data.

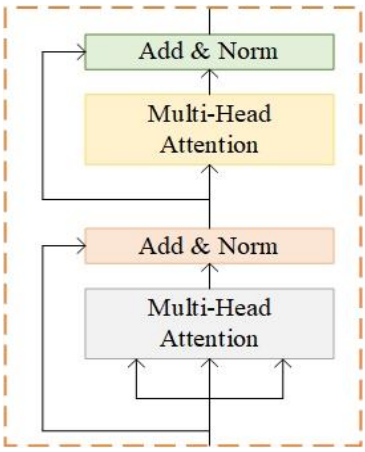

Figure 4. Basic Unit of Transformer

The Transformer was originally designed for natural language processing, requiring both an encoder and a decoder. However, in this study, the encoder part is utilized for PV data imputation, with the output layer replacing the decoder using a fully connected layer.

3. Case Study

A. Dataset Description

The dataset used in this study is the Desert Knowledge Solar Centre dataset from Australia, covering the period from 2015-01-01 00:00:00 to 2015-12-31 23:55:00, with a time resolution of 5 minutes. It contains data from 27 distributed PV stations. The original dataset includes meteorological information such as irradiance, temperature, and humidity. However, considering the practical lack of meteorological data, this study focuses solely on the historical output data of the stations. The dataset is divided into training and testing sets in an 8:2 ratio. During the model training and testing process, data missing from only one station is considered.

B. Construction of Complete-Missing Data Pairs

Data imputation is an unsupervised problem, meaning that there is no corresponding complete data for the missing data, which poses challenges for model training. To validate the proposed method, this study adopts a manual approach to construct complete-missing data pairs, as shown in Equation (11), where D_{miss} represents the missing data, D_{complete} represents the complete data, $\odot$ denotes the Hadamard product, and M represents the mask matrix. In the mask matrix, 1 indicates that the data is known, and 0 indicates that the data is unknown. Each sample has a time span of one day, consisting of 288 data points. When constructing the mask matrix M , the missing positions are randomly determined, and the missing rate is set to 50%, meaning that 144 data points are missing in each sample.

$$D_{\text{miss}} = D_{\text{complete}} \odot M \quad (11)$$

C. Correlation Measurement of Distributed PV Stations

There are various metrics available to measure the correlation between distributed PV stations, such as mutual information, covariance, Euclidean distance, and Spearman's rank correlation coefficient [34,35]. In this study, the Pearson correlation coefficient is used to quantify the correlation between distributed PV stations. The calculation formula is as follows:

$$\rho_{x,y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} \quad (12)$$

where $\mathrm{cov}(X,Y)$ is the covariance between X and Y, σ_X and σ_Y are the standard deviations of X and Y, respectively. Figure 5 illustrates the correlation among 26 distributed photovoltaic sites. As can be seen from the Figure 5, most of the sites exhibit strong correlations, which is beneficial for data restoration.

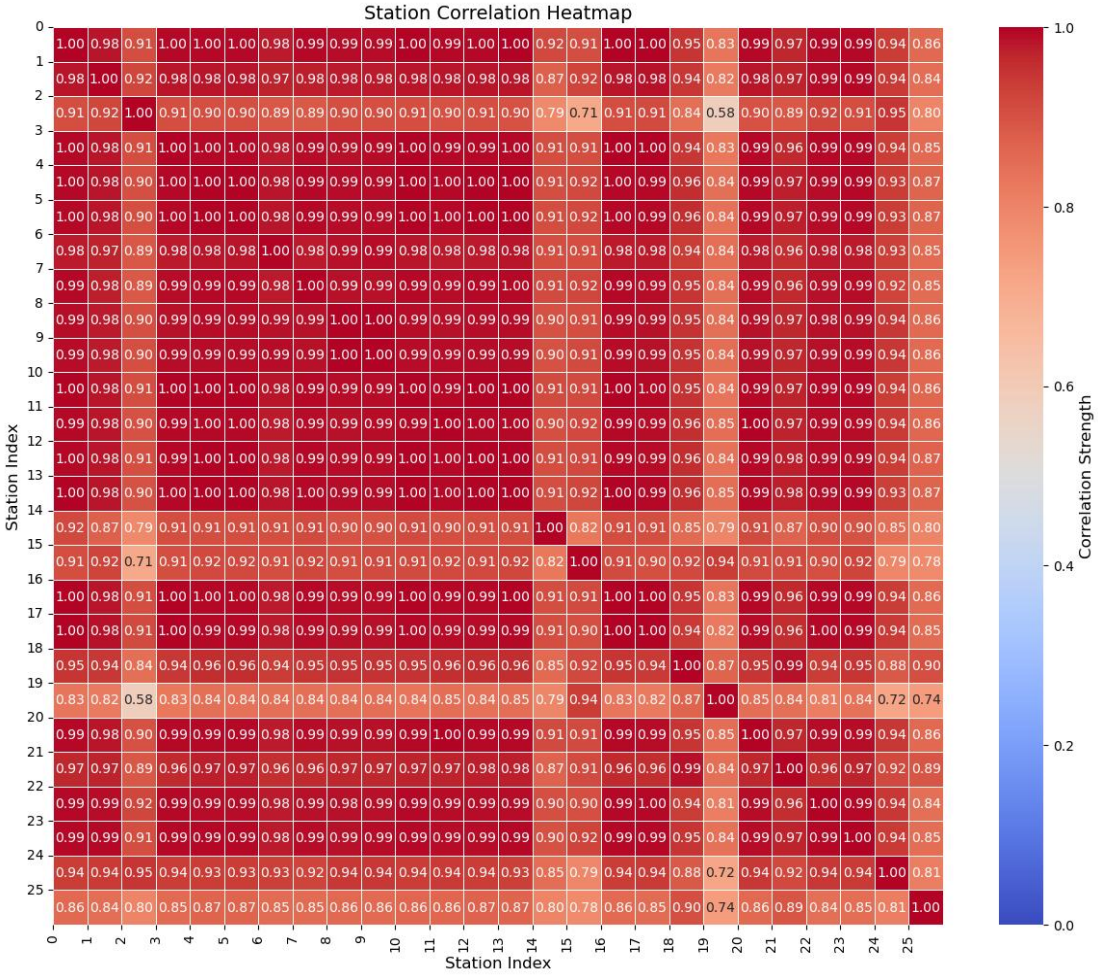

Figure 5. Station Correlation Heatmap

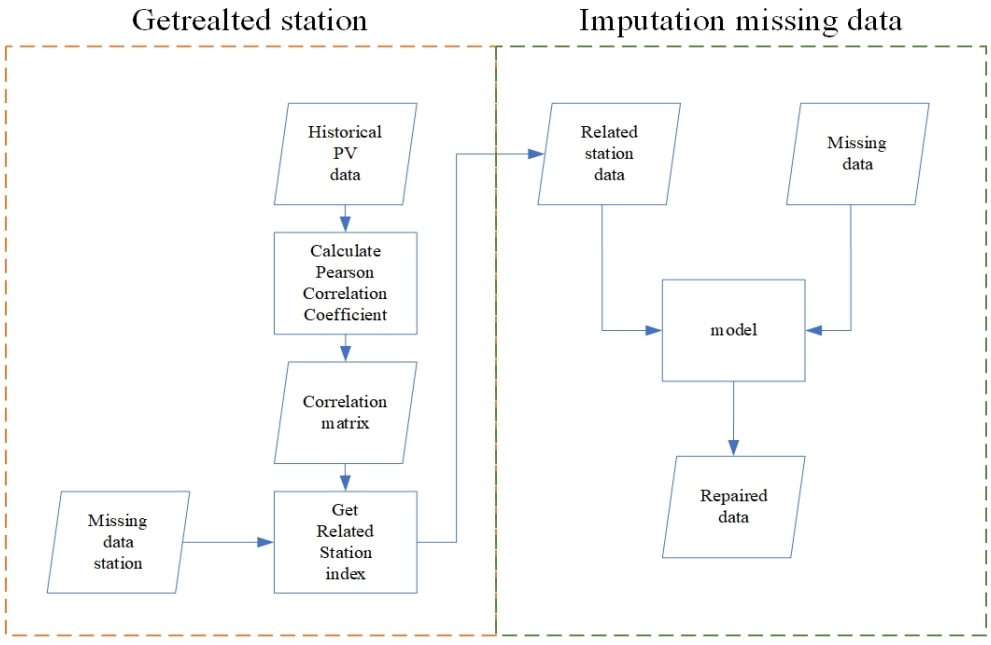

Figure 6. Imputation Process

D. Imputation Process and Evaluation Metrics

The imputation process proposed in this study is shown in Figure 6. First, the Pearson correlation coefficient is calculated based on the complete historical output data to obtain the correlation matrix. Second, according to the correlation matrix, several complete PV stations that are most correlated with the station with missing data are selected. In this study, two correlated stations are chosen. Finally, the complete data D_{related} from the correlated stations and the missing data D_{miss} from the station with missing values are jointly used as input to the neural network to obtain the imputed data D_{repaired} . The final imputation result $\hat{D}_{\text{repaired}}$ is obtained as shown in (13):

$$\hat{D}_{\text{repaired}} = D_{\text{repaired}} \odot (1 - M) + D_{\text{miss}} \quad (13)$$

To assess the imputation accuracy, we utilize Root Mean Squared Error (*RMSE*), Mean Absolute Error (*MAE*), and Mean Squared Error (*MSE*) as evaluation metrics. The overall percentage improvement in accuracy is then obtained by considering all evaluation metrics. The calculation formulas are as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{D}_{\text{repaired}_i} - D_{\text{complete}_i}\right)^2} \quad (14)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{D}_{\text{repaired}_i} - D_{\text{complete}_i}\right| \quad (15)$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{D}_{\text{repaired}_i} - D_{\text{complete}_i}\right)^2 \quad (16)$$

$$\text{ratio} = \left(1 - \frac{RMSE_{\text{related}} + MAE_{\text{related}} + MSE_{\text{related}}}{RMSE + MAE + MSE}\right) \cdot 100\% \quad (17)$$

E. Comparison of Results

Figure 7 illustrates the results of data imputation using the GRU model. It can be observed that the imputation performance, when both the output data from the correlated stations and the missing data from the target station are input into the model, is superior to the imputation accuracy obtained without incorporating the data from the correlated stations.

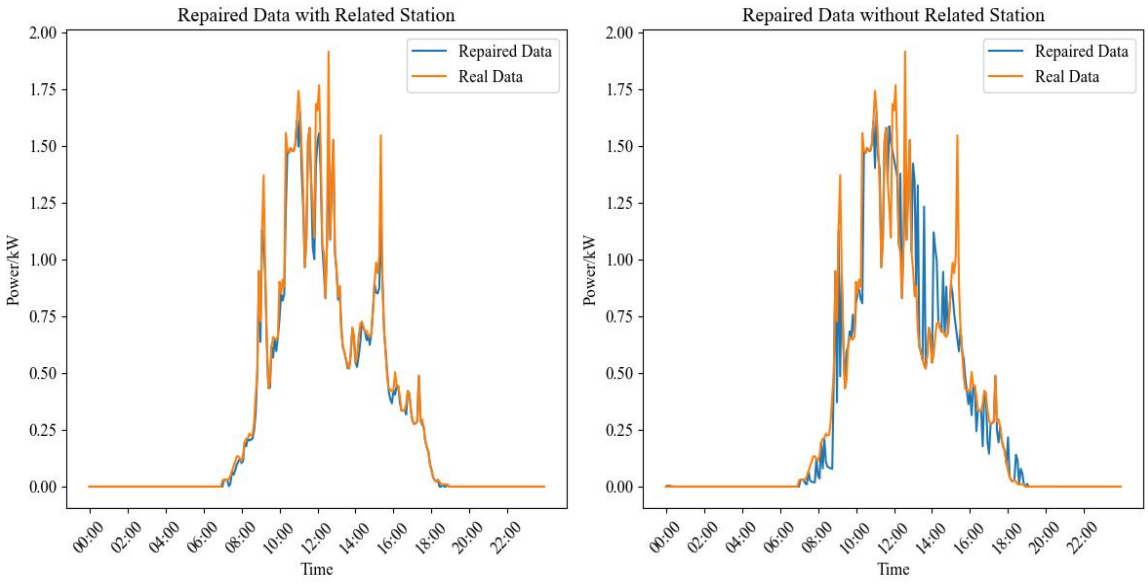

Figure 7. Imputation Results using GRU

Table 1: Imputation Performance of Different Models

Model	*RMSE*	*MAE*	*MSE*	ratio
GRU/GRU_NO	0.0460/0.1037	0.0199/0.0417	0.0026/0.0116	55.82%
LSTM/LSTM_NO	0.0544/0.0739	0.0218/0.0292	0.0034/0.0068	27.53%
TCN/TCN_NO	0.0442/0.1521	0.0198/0.0569	0.0024/ 0.0239	71.50%
Transformer/Transformer_NO	0.0722/0.0825	0.0401/0.0325	0.0062/0.0096	4.90%

The quantitative comparison of imputation accuracy is shown in Table 1. "Model_NO" represents the model where no data from correlated stations is used as input. As can be observed, the evaluation metrics for GRU, LSTM, and TCN all indicate that incorporating data from correlated stations significantly improves the imputation accuracy, with a notable increase. The imputation accuracy for GRU and TCN models improves by more

than 50%.

It is noteworthy that Transformer did not exhibit significant improvement, and in comparison to other models, its imputation accuracy was even lower. We hypothesize that this is due to the fact that the attention mechanism in Transformer is more suited for modeling global information, whereas PV data imputation requires a more focused consideration of local temporal dependencies. Furthermore, compared to the other three models, Transformer has a larger number of parameters, which requires more training data to adequately capture features. Given that the dataset used in this study is relatively small, Transformer was unable to fully learn the underlying data characteristics.

4. Conclusion

This paper focuses on data imputation using only historical output data from distributed PV stations. Due to the strong correlation between PV output and meteorological conditions, neighboring stations often experience similar environmental factors, leading to comparable output fluctuations. This study demonstrates that leveraging correlated station data can effectively support missing data imputation.

Experimental results show that incorporating output data from correlated stations significantly improves imputation accuracy across multiple neural network models. Notably, the GRU and TCN models exhibit over 50% accuracy improvement, confirming the effectiveness of this approach.

5. Future Work

Despite its advantages, the proposed method has limitations. In cases where extreme weather conditions cause widespread data loss across multiple PV stations in a region, correlated station data may also be unavailable, reducing the effectiveness of this approach. Future research should explore alternative strategies, such as hybrid imputation methods integrating external meteorological forecasts, probabilistic models, or GANs to enhance robustness under extreme conditions. Additionally, expanding the dataset and testing on a broader range of PV systems could further validate the generalizability of this method.

References

[1] R. Ahmed, V. Sreeram, Y. Mishra, M.D. Arif. A Review Evaluation of the State-of-the-Art in PV Solar Power Forecasting: Techniques Optimization. Renewable and Sustainable Energy Reviews, 2020, 124, 109792. DOI: 10.1016/j.rser.2020.109792

[2] U.K. Das, K.S. Tey, M. Seyedmahmoudian, S. Mekhilef, M.Y.I. Idris, et al. Forecasting of Photovoltaic Power Generation Model Optimization: A Review. Renewable and Sustainable Energy Reviews, 2018, 91, 912-928. DOI: 10.1016/j.rser.2017.08.017

[3] I. Pratama, A.E. Permanasari, I. Ardiyanto, R. Indrayani. A Review of Missing Values Handling Methods on Time-Series Data. 2016 International Conference on Information Technology Systems and Innovation (ICITSI), 2016, 1-6. DOI: 10.1109/ICITSI.2016.7858189

[4] H. De Silva, A. Shehan Perera. Missing Data Imputation Using Evolutionary k-Nearest Neighbor Algorithm for Gene Expression Data. 2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer), 2016, 141-146. DOI: 10.1109/ICTER.2016.7829911

[5] F. Camastra, V. Capone, A. Ciaramella, A. Riccio, A. Staiano, et al. Prediction of Environmental Missing Data Time Series by Support Vector Machine Regression Correlation Dimension Estimation. Environmental Modelling & Software, 2022, 150, 105343. DOI: 10.1016/j.envsoft.2022.105343

[6] F. Tang, H. Ishwaran. Random Forest Missing Data Algorithms. Statistical Analysis and Data Mining: An ASA Data Science Journal, 2017, 10(6), 363-377. DOI: 10.1002/sam.11348

[7] V. Ravi, M. Krishna. A New Online Data Imputation Method Based on General Regression Auto Associative Neural Network. Neurocomputing, 2014, 138, 106-113. DOI: 10.1016/j.neucom.2014.02.037

[8] Z.J. Sun, L. Xue, Y.M. Xu, Z. Wang. A Review of Deep Learning Research. Application Research of Computers, 2012, 29(08), 2806-2810. DOI: 10.3969/j.issn.1001-3695.2012.08.002

[9] R.Y. Liao, Y.B. Liu, Y.D. Shen, H.J. Gao, D.L. Tang, et al. A Coupling Enhancement Method for Time-Series Data of Distributed Photovoltaic Clusters Based on Bidirectional Recurrent Interpolation Network. Power System Technology, 2024, 18(7), 2784-2794. DOI: 10.13335/j.1000-3673.pst.2-023.1681

[10] W.J. Zhang, Y.H. Luo, Y. Zhang, D. Srinivasan. SolarGAN: Multivariate Solar Data Imputation Using Generative Adversarial Network. IEEE Transactions on Sustainable Energy, 2020, 12(1), 743-746. DOI: 10.1109/TSTE.2020.3004751

[11] D.T. Neves, M.G. Naik, A. Proença. SGAIN, WSGAIN-CP, WSGAIN-GP: Novel GAN Methods for Missing Data Imputation. International Conference on Computational Science. Cham: Springer International Publishing, 2021, 98-113. DOI : 10.1007/978-3-030-77961-0_10

[12] K.Y. Liu, F.Z. Zhou, H. Zhou. Distribution Network Substation Data Restoration Based on Time-Series Signal Image Encoding Generative Adversarial Network. Power System Protection Control, 2022, 50(24), 129-136. DOI: 10.19783/j.cnki.pspc.2-20256

[13] X.J. Qu, Z. Liu, C.Q. Wu, A.Q. Hou, X.Y. Yin, et al. MFGAN: Multimodal Fusion for Industrial Anomaly Detection Using Attention-Based Autoencoder Generative Adversarial Network. Sensors. 2024, 24(2), 637. DOI:10.3390/s24020637

[14] E. Oh, T. Kim, Y.H. Ji, S. hyalia. STING: Self-Attention Based Time-Series Imputation Networks Using GAN. 2021 IEEE International Conference on Data Mining (ICDM), 2021, 1264-1269. DOI: 10.1109/ICDM51629.2021.00155

[15] Y. Zhang, B.H. Zhou, X.R. Cai, W.Y. Guo, X.K. Ding, et al. Missing Value Imputation in Multivariate Time Series with End-to-End Generative Adversarial Networks. Information Sciences, 2021, 551, 67-82. DOI: 10.1016/j.-ins.2020.11.035

[16] L.F. Xu, L.Y. Xu, J. Yu. Time Series Imputation with GAN Inversion Decay Connection. Information Sciences, 2023, 643, 119234. DOI: 10.1016/j.ins.202-3.119234

[17] Q.T. Phan, Y.K. Wu, Q.D. Phan, H. Lo. A study on

missing data imputation methods for improving hourly solar dataset. 2022 8th International Conference on Applied System Innovation(ICASI), 2022, 21-24. DOI: 10.1109/ICASI55125.2022.9774453

[18] M. Shen, H.Z. Zhang, Y.X. Cao, F. Yang, Y.G. Wen. Missing data imputation for solar yield prediction using temporal multi-modal variational auto-encoder. Proceedings of the 29th ACM International Conference on Multimedia. 2021, 2558 - 2566. DOI: 10.1145/34740-85.3475430

[19] X.Y. Zhang, M. Wen, J.X. Li, H.Y. Huang. Distributed PV Power Forecasting Considering Spatio-Temporal Dependence. International Conference on Smart Grid Smart Cities (ICSGSC), 2023, 487-493. DOI: 10.1109/ICSGSC59580.2023.10319128

[20] S. Hochreiter, J. Schmidhuber. Long Short-Term Memory. Neural Computation, 1997, 9(8), 1735-1780. DOI: 10.1162/neco.1997.9.8.1735

[21] J. Ma, J.C.P. Cheng, F.F. Jiang, W.W. Chen, M.Z. Wang, et al. A Bi-Directional Missing Data Imputation Scheme Based on LSTM Transfer Learning for Building Energy Data. Energy and Buildings, 2020, 216, 109941. DOI: 10.1016/j.enbuild.2020.109941

[22] D. Li, L.H. Li, X.L. Li, Z.W. Ke, Q.H. Hu. Smoothed LSTM-AE: A Spatio-Temporal Deep Model for Multiple Time-Series Missing Imputation. Neurocomputing, 2020, 411, 351-363. DOI: 10.1016/j.ne-ucom.2020.05.033

[23] N. Li, F.X. He, W.T. Ma, L. Jiang, X.P. Zhang. State of Charge Estimation of Lithium-Ion Batteries Based on Empirical Mode Decomposition Gated Recurrent Unit Neural Network. Transactions of China Electrotechnical Society, 2022, 37(17), 4528-4536. DOI: 10.19595/j.cnki.1000-6753.tces.211069

[24] Z.P. Zhang, Y.Q. Zhang, A. Zeng, D. Pan, Y.Z. Ji, et al. Time-Series Data Imputation via Realistic Masking-Guided Tri-Attention Bi-GRU. ECAI, 2023, 3074-3082. DOI: 10.3233/FAIA230625

[25] Q.X. Tan, M. Ye, B.Y. Yang, S.Q. Liu, A.J. Ma, et al. Data-GRU: Dual-Attention Time-Aware Gated Recurrent Unit for Irregular Multivariate Time Series. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(1), 930-937. DOI: 10.1609/aaai.v34i01.5440

[26] Q.T. Li, Y. Xu. VS-GRU: A Variable Sensitive Gated Recurrent Neural Network for Multivariate Time Series with Massive Missing Values. Applied Sciences, 2019, 9(15), 3041. DOI: 10.3390/app9153041

[27] L. Zhang, G.F. Ren, S.L. Li, J.S. Du, D.Y. Xu, et al. A novel soft sensor approach for industrial quality prediction based TCN with spatial temporal attention. Chemometrics Intelligent Laboratory Systems, 2025, 257, 105272. DOI: 10.1016/j.chemolab.2024.105272

[28] Y.F. Mao, M. Yang, P. Li, Z.J. Ou. A Missing Data Imputation Method for Electricity Consumption Data Based on TCN-Attention with Mask Tokens. 2024 4th International Conference on Consumer Electronics Computer Engineering (ICCECE), 2024, 513-517. DOI: 10.1109/ICCECE61317.2024.10504227

[29] K.K.R. Samal. Auto Imputation Enabled Deep Temporal Convolutional Network (TCN) Model for PM2.5 Forecasting. EAI Endorsed Transactions on Scalable Information Systems, 2025, 12(1). DOI: 10.4108/eetsis.5102

[30] Z. Luo, Y.H. Wu, J.X. Zhu, W.J. Zhao, G. Wang, et al. Ultra-Short-Term Wind Power Forecasting Based on a Multi-Scale Time Series Block Autoencoder Transformer Neural Network Model. Power System Technology, 2023, 47(09), 3527-3537. DOI: 10.1109/ACCESS.2024.337-3798

[31] J. Liu, S. Pasumarthi, B. Duffy, E. Gong, K. Datta, et al. One Model to Synthesize Them All: Multi-Contrast Multi-Scale Transformer for Missing Data Imputation. IEEE Transactions on Medical Imaging, 2023, 49(2), 2577-2591. DOI: 10.1109/TMI.2023.3261707

[32] A. Lotfipoor, S. Patidar, D.P. Jenkins. Transformer Network for Data Imputation in Electricity DemData. Energy and Buildings, 2023, 300, 113675. DOI: 10.1016/j.enbuild.2023.113675

[33] A. Yarkın Yıldız, E. Koç, A. Koç. Multivariate Time Series Imputation with Transformers. IEEE Signal Processing Letters, 2022, 29, 2517-2521. DOI: 10.1109/LSP.2022.3224880

[34] J. Benesty, J.D. Chen, Y.T. Huang, I. Cohen. Pearson Correlation Coefficient. Noise Reduction in Speech Processing, 2009, 1-4. DOI: 10.1007/978-3-642-00296-0_5

[35] J. Adler, I. Parmryd. Quantifying Colocalization by Correlation: The Pearson Correlation Coefficient is Superior to the Mander's Overlap Coefficient. Cytometry Part A, 2010, 77(8), 733-742. DOI: 1-0.1002/cyto.a.20896