Imputation as Service Using Support Vector Regression: Application to a Photovoltaic System in Algeria

Mohamed Taki Eddine Seddik, Ouahab Kadri, Mohamed Rida Abdessemed

Proceedings of 1st National Conference of Materials sciences And Engine ering, (MSE'22)

MSE'22

IMPUTATION AS SERVICE USING SUPPORT VECTOR REGRESSION: APPLICATION TO A PHOTOVOLTAIC SYSTEM IN ALGERIA

Mohamed Taki Eddine Seddik*[1], Ouahab Kadri[2] and Mohamed Rida Abdessemed[3]

[1] Lastic Laboratory, Department of Computer Science, University of Batna 2, Algeria
[2] LAP Laboratory, Department of Computer Science, University of Batna 2, Algeria
[3] Lastic Laboratory, Department of Computer Science, University of Batna 2, Algeria

*correspondence E-mail: seddikmedtakieddine@gmail.com.

Keywords:
Missing value
Imputation
Classification
Photovoltaic system
Meteorological data

ABSTRACT

This paper aims to test the most common imputation methods' effectiveness and choose the most appropriate methods for our data model. In the experimental study, we applied imputation to missing data using the imputation methods: fFill, bFfil, Drop, and Support Vector Regression (SVR). An easy and practical means of comparison is used to evaluate the effectiveness of imputation methods. Therefore, the classification quality criterion is used, and column reference graphs are used because they have a statistically significant relationship. The SVR imputation method was very reliable, and it helped us make a reasonable classification.

1. Introduction

Every photovoltaic (PV) system installation is based on two essential elements: the physical equipment and the meteorological data. The quality of the prognosis of the PV system depends on meteorological data. These data may include missing values for one or many reasons. Despite the growing amount of data available and the emergence of Big Data, the problem of missing data remain widespread in meteorological data. The presence of missing data can impact different levels in terms of internal validity, the validity of associations between variables, and the generalization of an association between two variables. Ignoring missing data can lead to loss of accuracy; the most effective way to solve this problem is to impute the missing values.

In industries, the issue of missing values is common. This problem has several causes: acquiring data by non-automatic methods, defective sensors or components, and incorrect measurements. Missing data may not always arise by sheer chance. We can distinguish three categories: Missing completely at random (MCAR), Missing at random (MAR), and missing not at random (MNAR)(Kadri et al., 2017). Missing data may threaten internal validity at different points in the research process: at the time of sample selection, random assignment to the experimental or control group, collection (complete or partial non-response, attrition), and statistical analysis of the data. The mathematical study critiques each missing data processing method according to the distribution characteristics of missing data and their process of existence. Among the methods known as "simple" are the deductive method and the cold-deck method, the analysis methods of complete or available subjects, the methods of prediction by regression, and the indicator method (Zhang et al., 2017).

One of the most common ways to deal with missing values is to ignore the observations that include them. (Beliakov et al., 2017). The complete-case analysis approach focuses only on subjects without missing data. On the other hand, The available-case analysis approach employs the maximum number of complete instances for individual parameter estimates. In the case of the indicator method, each variable with missing data is associated with its missing data indicator variable in the descriptive term of the final regression model explaining the end-point variable. The methods with weighted estimation equations use modeling of the existing process of the missing data to assign weights to the covariates for the regression analysis of the end-point variable.

With simple imputation, each missing data is replaced by predicted or simulated data, and the analysis will cover all records. There are many simple imputation methods in the literature; we will give some examples to explain more what is simple imputation. First, Simple imputation using the last observation is characterized by replacing all the missing values with the observed values average of the same variable or unbiased estimates if the data is MCAR(Zhang, 2016). Second, Cold-Deck and Hot-Deck imputation, the Cold-Deck is a method that uses a similar dataset(another survey) to impute the missing values. However, Hot-Deck uses similar individuals from the same dataset. Many well-known branches come from Hot-deck, such as The random Hot-Deck, the hierarchical sequential Hot-Deck, and the metric Hot-Deck.

Third, Simple imputation by average, this method is based on assigning a value to incomplete observations of a variable. Its implementation does not require the

availability of auxiliary variables relevant to the analysis of the missing variable because the average of the values given by the respondents replaces it. Analytically, when an individual does not answer a question in a survey and does not give a value, we calculate the average to impute it. Forth, Simple imputation by a regression model(Blackwell et al., 2017). This method is characterized by replacing a missing value Yi with a predicted value Y* obtained by regression of Y on X1, X2.

There is another very well-known method called multiple imputations. As its name implies, It entails repeatedly imputing missing data to aggregate the findings and minimize the error (noise) caused by imputation. It also helps to define a measure of the uncertainty caused by completion. Maintained the original data variability by creating imputed values based on variables correlated with missing data and absence causes. Uncertainty is taken into account by creating different versions of missing data and observing the variability between the imputed datasets. Despite the variety of choices in this field, there is little effort to rectify the missing data in photovoltaic systems. We quote here the most relevant according to our point of view:

Ioannis and Panapakidis have proposed a novel methodology for incomplete data completion. Their methodology uses the clustering tool to group the available data patterns into homogeneous clusters. The methodology is not dependent on data size, data resolution, and the amount of missing and incomplete data. It can be used for virtually any type of time series(Panapakidis & Dagoumas, 2016).

Haydar and Zoe have compared the accuracy of 36 imputation methods for solar irradiance series over a real dataset recorded in Australia under 16 experimental conditions. The experiments are run in a semi-Monte Carlo setting, in which missing values are randomly generated in the solar irradiance series(Demirhan & Renwick, 2018).

Tahasin et al. have proposed an iterative MTL-GP-TS model that learns/imputes unobserved or missing values in a time series dataset associated with the solar panel of interest to predict the PV trend. Additionally, the method improves and generalizes the traditional multi-task learning for Gaussian Process to learn global trends and locally irregular components in time series(Shireen et al., 2018).

The remainder of this paper is structured as follows: Section 2 contains a brief description of the background of the photovoltaic system. Section 3 introduces the Pvlib-python toolbox. Section 4 presents the proposed methodology, Experimental design and the experimental results are presented in section 5. Section 6 provides the conclusion.

2. Photovoltaic System

A Photovoltaic (PV) system or solar energy system is a power system designed to provide usable solar energy using photovoltaic energy. It consists of a multi-component installation consisting of solar panels to absorb and convert sunlight into electricity, a solar inverter to switch from direct current to alternating current, mounting, wiring, and other electrical accessories for installing a working system(figure 1 illustrates the important components). It can also use a solar tracking system to improve the overall system performance and include an integrated battery solution, as the prices of storage devices should decrease. A solar network includes only the solar panels, the visible part of the PV system, and does not include all the other materials, often summarized as a system balance (BOS)(Baumgartner, 2017). In addition, photovoltaic systems directly convert light into electricity and should not be confused with other technologies, such as concentrated solar energy or solar thermal energy used for heating and cooling.

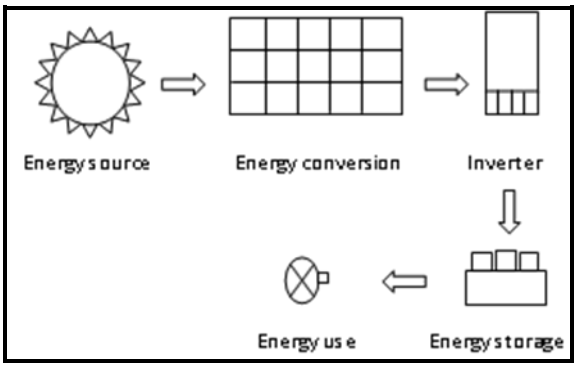

Figure 1 Components of a photovoltaic system.

We can distinguish three categories of Photovoltaic systems: residential roofs, commercial roofs, and ground service systems. Their capacities range from a few kilowatts to hundreds of megawatts. A typical residential system is about 10 kilowatts mounted on an inclined roof. In contrast, commercial systems can reach a megawatt scale and are typically installed on low-pitched or even flat roofs. Although rooftop systems are small and cost per watt higher than larger utility systems, they represent the largest share of the market. However, there is a growing trend toward larger-scale power plants, particularly in the "sunshine belt" region (Ogbeche, 2018).

An isolated PV system supplies the user with electricity without connecting to the electricity grid. Using a PV system is often the only way to electrify when unavailable network current, such as houses on isolated sites, islands, and mountains(Lazar et al., 2018). This type of system requires the use of batteries for the storage of electricity and a charge controller to ensure the durability of the batteries.

A grid-connected photovoltaic system is directly connected to the grid using an inverter. This system is extremely simple for the producer/consumer since it is the responsible grid balancing electricity production and consumption. In the case of systems connected to the network, it is imperative to convert the direct current produced by the photovoltaic system into an alternating current synchronized with the network. An inverter is used to perform this conversion. The typical efficiency of an inverter is about 95%. Different powers and inverters are designed specifically for photovoltaic applications(Ramli et al., 2017). The inverter also has a network decoupling function that prevents power from being injected into the grid when it is not in operation and an overvoltage protection function.

The components of a photovoltaic system depend on the type of application. In an isolated site, the main components are polar panels, DC / DC controller/charger, a storage system (batteries, capacitors ...), possibly an inverter if the consumption is powered by alternating current, and possibly an auxiliary generator. In a grid-connected photovoltaic system the main components are polar panels (+ support structure), DEC cut-off and DC protection case-inverters, AC protection, and a cut-off cabinet(Pereira et al., 2017).

A photovoltaic system is a long-term investment since its lifespan is generally over three decades and can even reach four decades. Thus, It should be noted that most module manufacturers offer a guarantee of performance over 20 to 25 years. In general, they guarantee that the module will always have 80% of its peak power after this time. The mechanical guarantee (resistance to hail, transport, and others) is generally between 2 and 5 years(Kong et al., 2017). For the other components of a photovoltaic system, the inverters will have a 10-15 years lifespan, and the batteries will have to be replaced every ten years(Akinyele et al., 2017).

3. SOFTWARE TOOLS

To develop our imputation system, we have used two toolboxes which are LIBSVM and PVLIB, using Python. Python is a high-level, structured, and open-source scripting language. It is multi-paradigm and multi-purpose. Originally developed by Guido van Rossum in 1989, it is, like most applications and open source tools (vanRossum, 1995).

C.Chang developed LIBSVM in 2001. An SVM toolbox includes classification approaches (C-SVC and nu-SVC) and regression approaches (epsilon-SVR, nu-SVR, and SVM for a class). It allows multi-class classification, and Its basal algorithm simplifies Platt's SMO algorithm and Joachims' SVMLight algorithm. LIBSVM offers a very simple interface to use the different possible configurations. It comprises two essential modules: Svmtrain for learning and Svmpredict for classification.

PVLIB Python is a toolbox that offers several functions and classes to model photovoltaic systems(Holmgren et al., 2015). PVLIB Python was developed from the PVLIB MATLAB toolkit developed by Sandia National Laboratories and implemented many models and methods developed at L'ABS. The main mission of Pvlib-python is to provide open, reliable, interoperable, and reference implementations of PV system models. The Pvlib-python toolbox is a well-tested code of procedure that implements models of PV systems. Pvlib-python also provides a collection of classes for users who prefer object-oriented programming. These classes can help users organize tracking data, provide "smart" functions with more flexible inputs, and simplify the modeling process for common architectures. Classes do not add any algorithm beyond what is available in the procedure code, and most object methods are simple packages around the corresponding procedural code.

The platform used to implement our application is Google Colab. It is a free cloud version of Jupyter Notebook. Google Colab allows users to use Machine Learning tools directly in the cloud using virtual machines based on TPUs and GPUs.

4. Proposed approach

This section will deal with missing data using the most common imputation methods. The principle of these methods is to generate values to estimate the missing data.

We tested the utility of imputation on data classification. We predicted class labels. The observations and associated class labels are divided into two sets, the training set and the testing set. The observations(the training set) that make up the learning are randomly sampled from the analyzed data. The remaining data(testing set) will not be used to build the classifier.

The testing set is used to estimate the predictive accuracy of a classifier. The accuracy of a classifier is the percentage of test observations that are correctly classified by the classifier. As we said at the beginning of this article, this work aims to test the efficiency of the imputation methods available in Python language and choose the best between them.

With Numpy and Matplotlib, the Pandas library is one of the core libraries for Python data science. It provides powerful and easy-to-use data structures and methods to simplify the exploitation of these structures. (McKinney, 2010). Scikit-learn is a Python toolbox dedicated to machine learning. It includes functions to estimate random forests, logistic regressions, algorithms of classification, and machinery with support vectors. It is designed to harmonize with others Python components, especially NumPy and SciPy(Pedregosa et al., 2011).

Iris is the most well-known database. It is found almost in all the tutorials on the net (Bailenson et al., 2008). This dataset includes 150 observations equally distributed among the three species of iris flowers (Setosa, Versicolor, and Virginica). Four characteristics are measured for each observation (i.e., the length and width of the sepal and petal, in centimetres).

To apply data imputation, we have created artificial tables containing missing values. For example, we deleted 5% of the data in a file. The information concerned by this deletion is as follows: Sepal-length 4, Sepal-width 9, Petal-length 4, and Petal-width 11. We have also generated other tables but with different suppression rates. Moreover, we added 10% of the data to another file. The data corresponding to this operation is as follows: Sepal-length 9, Sepal-width 11, Petal-length 5, and Petal-width 16.

We used four methods of imputation. The first method (M1), fFill(forward fill), imputes missing values by filling them with the preceding column or row value. In contrast, the second method (M2) uses the next observed value to fill in the missing value, known as Bfil (backward fill). The third method (M3) drop is to remove records that have missing values. In the fourth method, SVR(M4), we use Kernel interpolation to fill in the missing value. We ignored the index, and we treated the values as equally distributed.

First, we have created a prediction model based on the kNN (k nearest neighbor) classification method. We

performed a comparative study based on the results found. Table 1 shows the result predictions for the database iris with the imputed copies, respectively. According to these results, we concluded that Ffil is the worst method. On the other hand, the other methods have the same imputation quality.

Table 1 Prediction results of iris dataset.

Data	M1	M2	M3	M4
Original	0.98	0.98	0.98	0.98
MV with 5%	0.97	0.98	0.98	0.98
MV with 10%	0.94	0.97	0.97	0.97

We used the iris dataset in this section to show the impact of classification. However, in the next section, we will use Python's PVLIB to apply our proposed method to our study case(photovoltaic systems). Python's PVLIB provides functions and classes that make it easy to obtain weather forecast data and convert it to PV power. Users can extract standard weather forecast data relevant to PV power modeling from the NOAA / NCEP / NWS models (Gutman & Ignatov, 1998), including the GFS used in our case. The first step is importing the data with the Global Forecast System (GFS) module.

5. RESULT AND DISCUSSION

Data imported with the GFS model are data measured with units of the American system. However, our application has a different format (see table 2). Consequently, we have modified the structure of data observation before using it. The last action in this step is to save the data in CSV format using the Pandas. To evaluate the efficiency of the imputation method, we must have an easy and practical means of comparison.

Table 2 An Example of forecast results in 09-06-2018(12a.m, 3a.m and 6a.m).

Parameter	V1	V2	V3
index2018-06-09	00:00:00	03:00:00	06:00:00
Weather	19.66452	18.928528	21.584686
Wind speed	2.5121348	2.148635	2.361764
Ghi	0	0	29.65
Dni	0	0	28.61
Dhi	0	0	27.28
Total clouds	0	2	39
Low clouds	0	0	0
Mid clouds	0	2	39
High clouds	0	0	0

The following figures(2,3 and 4) show Algeria's forecast results obtained by GFS.

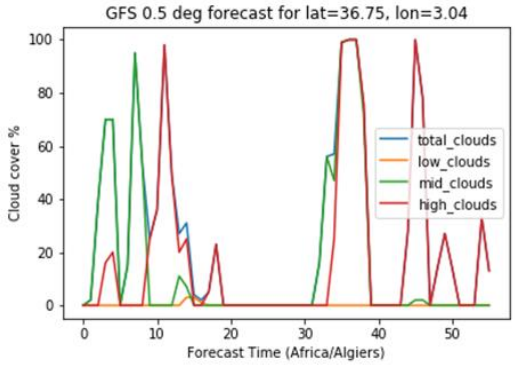

Figure 2 Total clouds, Low clouds, Mid clouds and High clouds.

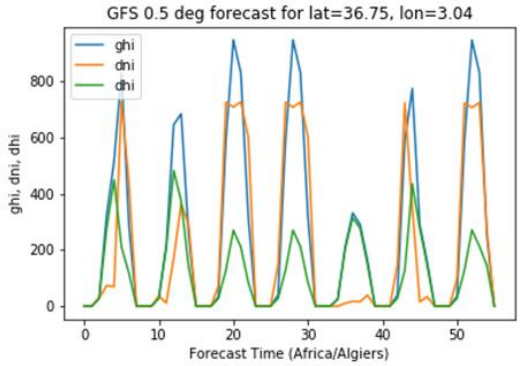

Figure 3 GHI, DN, DHI.

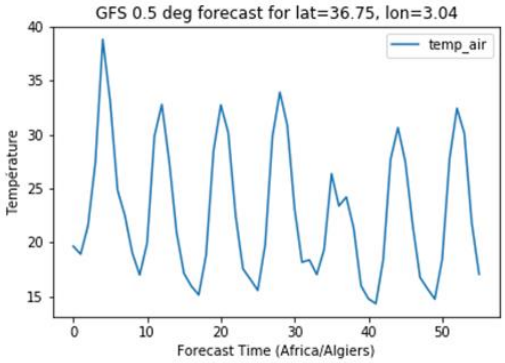

Figure 4 Prediction of Temperature.

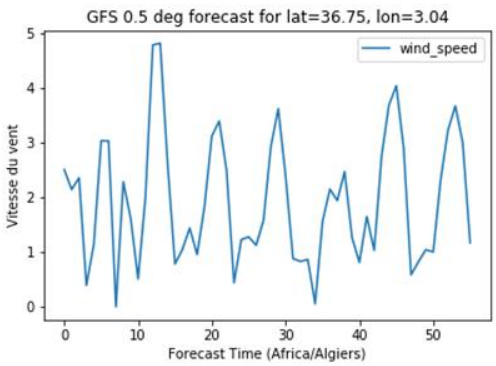

Figure 5 Wind speed.

To simulate files with missing data, we created firstly two copies of the origin.csv file. Then we deleted e 5%

of the first data file, and we removed 10% of the second data file.

The result of this operation is shown in figure 6.

	A	B	C	D	E	F	G	H	I	J
1	index	temp_air	wind_speed	ghi	dni	dhi	total_clouds	low_clouds	mid_clouds	high_clouds
2	2018-06-09 00:00:00+01:00	19.66452	2.5121348	0	0	0	0	0	0	0
3	2018-06-09 03:00:00+01:00	18.928528	2.148635	0	0	0	2	0	2	0
4	2018-06-09 06:00:00+01:00	21.584686	2.361764		28.6123825	27.2856253	39	0	39	0
5	2018-06-09 09:00:00+01:00	27.48233	0.395888	322.143855	73.2091901	275.465732	70	0	70	16
6	2018-06-09 12:00:00+01:00	38.845276	1.14067	516.283999		449.915597	70	0	70	20
7	2018-06-09 15:00:00+01:00	33.09372		828.925278	727.877165	209.859684	0	0	0	0
8	2018-06-09 18:00:00+01:00	24.86676	3.0364952	284.64433	437.49416	116.645714	15	0	15	0
9		22.483612	0.002152221	0	0	0	95	0		0
10	2018-06-10 00:00:00+01:00	19.045074	2.2887406	0	0	0	53	0	53	0
11	2018-06-10 03:00:00+01:00	16.999115	1.6214906	0	0	0	25	0	0	25

Figure 6 Data after deleting 5%.

We applied the four methods mentioned previously to the Pvlib data, as shown in Python code in figure 7.

```
import pandas as pd
import matplotlib.pyplot as plt

latitude, longitude, tz = 36.75, 3.04, 'Africa/Algiers'

damaged_data = pd.read_csv("db/damaged_data 5%.csv",sep=';')

imputation_dropna = damaged_data
imputation_ffile  = damaged_data
imputation_bfile  = damaged_data
imputation_intermplation = damaged_data
```

Figure 7 Code of imputation method.

The superposition tacitly compares the graphs' results using pictures (Photoshop) software show.

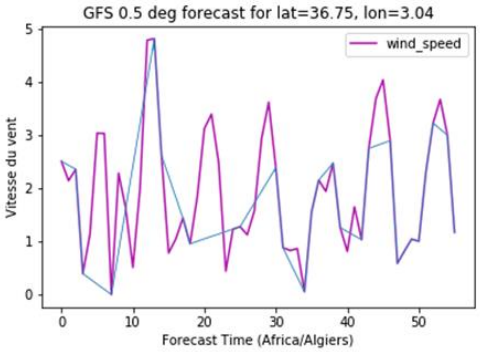

Figure 8 The blue graph describes Pvlib wind speed prediction, and the purple describes the prediction after imputation using fFill.

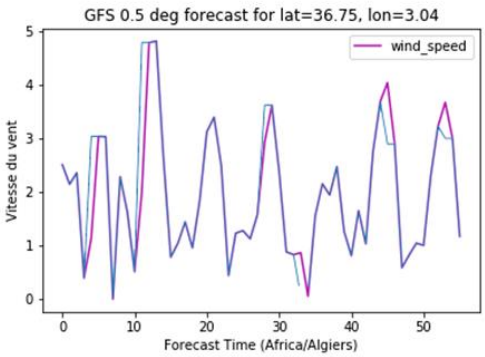

Figure 9 The blue graph describes Pvlib wind speed prediction, and the purple describes the prediction after imputation using Bfill.

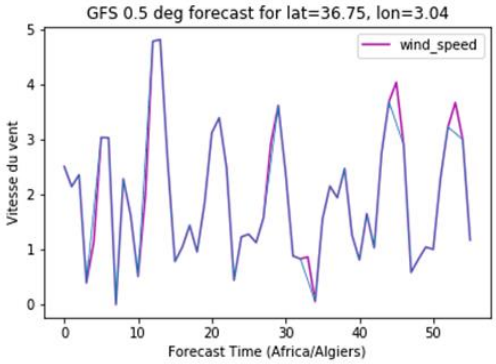

Figure 10 Figure 8 The blue graph describes Pvlib wind speed prediction, and the purple describes the prediction after using Drop.

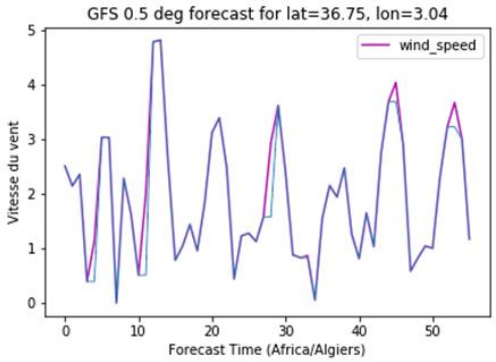

Figure 11 The blue graph describes Pvlib wind speed prediction, and the purple describes the prediction after imputation using the SVR method.

We performed a comparative study based on the results found. The four previous figures show the results of the wind speed predictions for the modified and origin database of Pvlib with the imputed copies, respectively. According to these results, we concluded that Ffilna is the worst method. On the other hand, the three other methods have the same imputation quality. It should be noted that the data nature does not greatly influence the imputation quality of methods, consequently, the classification rate or the prediction rate.

Based on the findings, we performed a comparative study. The outcomes of the predictions for the modified(imputed copies) and origin databases of Pvlib are shown in the four preceding figures. We concluded that Ffill is the weakest method. On the other hand, the rest approaches have the same imputation quality. In addition, It should be noted that the nature of the data has no impact on the imputation quality of methods nor the classification or prediction rates.

6. Conclusion

The present paper proposes several programming modules for incomplete data imputation. All photovoltaic installations must include sensors of physical quantities. These sensors are used to collect data to evaluate the performance of the photovoltaic system. However, the data may contain invalid values in practice, such as missing values. These circumstances can lead to an impossible system diagnosis. In addition, the neglect of

observations with missing values will incorrectly describe the system operation. Therefore, it will be better to impute incomplete data with intelligent technics. We tried to find the missing values of the incomplete data by using the imputation methods: Ffill, Bfil, Drop, and SVR, which were very reliable. This study represents the first step toward developing a diagnostic system that will be the objective of our future work.

ACKNOWLEDGEMENTS

The research has been generously supported by the Laboratory of Automation & Production Engineering and the National Agency for Development of Research and Technological Development Results [PRFU project: C00L07UN050220200003]. The authors would like to express their sincere appreciation for all the support provided.

7. References

Akinyele, D., Belikov, J., & Levron, Y. J. E. (2017). Battery storage technologies for electrical applications: Impact in stand-alone photovoltaic systems. *10*(11), 1760.

Bailenson, J. N., Pontikakis, E. D., Mauss, I. B., Gross, J. J., Jabon, M. E., Hutcherson, C. A., . . . John, O. J. I. j. o. h.-c. s. (2008). Real-time classification of evoked emotions using facial feature tracking and physiological responses. *66*(5), 303-317.

Baumgartner, F. (2017). Photovoltaic (PV) balance of system components: Basics, performance. In *The Performance of Photovoltaic (PV) Systems* (pp. 135-181). Elsevier.

Beliakov, G., Gómez, D., James, S., Montero, J., Rodríguez, J. T. J. F. S., & Systems. (2017). Approaches to learning strictly-stable weights for data with missing values. *325*, 97-113.

Blackwell, M., Honaker, J., King, G. J. S. M., & Research. (2017). A unified approach to measurement error and missing data: overview and applications. *46*(3), 303-341.

Demirhan, H., & Renwick, Z. J. A. E. (2018). Missing value imputation for short to mid-term horizontal solar irradiance data. *225*, 998-1012.

Gutman, G., & Ignatov, A. J. I. J. o. r. s. (1998). The derivation of the green vegetation fraction from NOAA/AVHRR data for use in numerical weather prediction models. *19*(8), 1533-1543.

Holmgren, W. F., Andrews, R. W., Lorenzo, A. T., & Stein, J. S. (2015). PVLIB python 2015. In 2015 ieee 42nd photovoltaic specialist conference (pvsc), the Hyatt Regency - New Orleans. pp. 1-5.

Kadri, O., Mouss, L., Abdelhadi, A. J. I. J. o. Q. E., & Technology. (2017). Fault diagnosis for a milk pasteurisation plant with missing data. *6*(3), 123-136.

Kong, F. K.-W., Tang, M.-C., Wong, Y.-C., Ng, M., Chan, M.-Y., & Yam, V. W.-W. J. J. o. t. A. C. S. (2017). Strategy for the realization of efficient solution-processable phosphorescent organic light-emitting devices: design and synthesis of bipolar alkynylplatinum (II) complexes. *139*(18), 6351-6362.

Lazar, E., Petreus, D., Etz, R., Patarau, T. J. A. i. E., & Engineering, C. (2018). Software Solution for a Renewable Energy Microgrid Emulator. *18*(1), 89-94.

McKinney, W. (2010). Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference, pp. 51-56.

Ogbeche, P. O. O. L. O. (2018). Empirical Estimation of Monthly Average Daily Solar Radiation and Solar Electricity Output from Sunshine Hours in Ogoja in Nigeria. *International Journal of Innovative Research and Development*, *7*(7), 5.

Panapakidis, I. P., & Dagoumas, A. S. J. A. E. (2016). Day-ahead electricity price forecasting via the application of artificial neural network based models. *172*, 132-151.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Dubourg, V. J. t. J. o. m. L. r. (2011). Scikit-learn: Machine learning in Python. *12*, 2825-2830.

Pereira, H. A., Freijedo, F. D., Silva, M., Mendes, V., Teodorescu, R. J. I. J. o. E. P., & Systems, E. (2017). Harmonic current prediction by impedance modeling of grid-tied inverters: A 1.4 MW PV plant case study. *93*, 30-38.

Ramli, M. A., Twaha, S., Ishaque, K., Al-Turki, Y. A. J. R., & Reviews, S. E. (2017). A review on maximum power point tracking for photovoltaic systems with and without shading conditions. *67*, 144-159.

Shireen, T., Shao, C., Wang, H., Li, J., Zhang, X., & Li, M. J. A. e. (2018). Iterative multi-task learning for time-series modeling of solar panel PV outputs. *212*, 654-662.

vanRossum, G. J. D. o. C. S. (1995). Python reference manual. (R 9525).

Zhang, Y., Alyass, A., Vanniyasingam, T., Sadeghirad, B., Flórez, I. D., Pichika, S. C., . . . Iljon, T. J. J. o. c. e. (2017). A systematic survey of the methods literature on the reporting quality and optimal methods of handling participants with missing outcome data for continuous outcomes in randomized controlled trials. *88*, 67-80.

Zhang, Z. J. A. o. t. m. (2016). Missing data imputation: focusing on single imputation. *4*(1).