

2020

2021

2022



PVAnalytics: A Python Package for Automated Processing of Solar Time Series Data

Kirsten Perry (NREL), William Vining (Sandia), Kevin Anderson (NREL), Matthew Muller (NREL), Cliff Hansen (Sandia)

PV Performance Modeling and Monitoring Workshop Salt Lake City, Aug 24, 2022

pvanalytics v0.1.2 **PVPerformance** MODELING COLLABORATIVE

pvanalytics v0.1.1

pvanalytics v0.1.0

https://github.com/pvlib/pvanalytics

Contents

1	PVAnalytics Background
2	Package Features
3	Algorithm Validation
4	Documentation Updates
5	Automated testing
6	Community growth
7	PVAnalytics v0.1.3 and Beyond

PVAnalytics Background

- Solar time series data can vary significantly in quality or lack critical metadata
- Several solar metrics dependent on data cleaning/filtering [1]
 - Performance loss rate (PLR)
 - Power production forecasting
 - Soiling loss
- PVAnalytics Python library: automated processing of solar time series data, including QA/QC
 - Data quality control and filtering
 - Identifying system characteristics, such as mounting configuration, tilt, and azimuth
 - Feature identification: clipping, day-night masking, clearsky detection
 - <u>https://pvanalytics.readthedocs.io/en/stable/</u>

[1] Lindig et. al. International collaboration framework for the calculation of performance loss rates: Data quality, benchmarks, and trends (towards a uniform methodology). Progress in Photovoltaics, 2021.

PVAnalytics Background (Continued)

- Design Principles behind PVAnalytics:
 - Open-source: tested, documented, and reusable
 - Independent of analysis workflow
 - Collection point for code which implements published algorithms
 - Collaboration between Sandia and NREL
 - Started as DuraMAT project: DOE-led consortium for PV module reliability and durability
 - Functions adapted from Solar Forecast Arbiter [1] and NREL PV Fleets Initiative [2]

[1] https://solarforecastarbiter-core.readthedocs.io/en/latest/

[2] D. Jordan et. al. *Photovoltaic fleet degradation insights*. Progress in Photovoltaics, 2022.





PLR distribution from the PV Fleets Initiative [2]

Package Features: Basic Time Series Filtering

Outlier detection and filtering: Hampel, Zscore, and Tukey filters

Stale data detection and filtering: Looks for consecutive repeating data

Interpolated data detection and filtering





Package Features: Advanced Time Series Filtering

Detecting missing data periods: Assign daily data a "completeness" score



[1] K. Perry, M. Muller. Automated Shift Detection in Sensor-Based PV Power and Irradiance Time Series. 2022 PVSC.

Data shift detection and filtering:

Uses changepoint detection to find massive, abrupt capacity changes. Described further in [1]



NREL | 6

Package Features: Feature Detection

- Day-night masking
 - Logic-based routine for masking day periods from night periods
- Clipping detection and filtering
 - Adapted from logic-based filter described in [1]
- Shading detection
 - Uses morphological image processing methods to identify shadows in GHI data [2]

[1] K. Perry, et. al. *Performance comparison of clipping detection techniques in AC power time series*. 2021 PVSC.

[2] Martin, C. E., Hansen, C. W., An Image Processing Algorithm to Identify Near-Field Shading in Irradiance Measurements, preprint 2016

Day-night masking on an AC power time series



Package Features: Irradiance Checks

Compare GHI sensor-based Irradiance quality checks: **Clearsky period** data to clearsky data. Filter consistency and physical filtering: Reno clearsky where GHI is within daily limits of GHI, DNI, and DHI method (1) insolation limit using QCrad criteria RMIS GHI RMIS GHI RMIS DHI 1000 Clearsky GHI ا اrradiance [W/m2] ج 000 00 RMIS DNI Within Daily Insolation Limit QCRAD Consistent 00 800 Irradiance (W/m^2) 00 600 00 Ital 400 00 Horizon 500 00 200 Global 100 00 0 04 02 03 05 06 02 Feb 2019 03 04 05 Feb 2019 15:00 18:00 21:00 06:00 09:00 12:00 00:00 03:00 20-Jan Date Date

[1] Reno, M.J. and C.W. Hansen, "Identification of periods of clear sky irradiance in time series of GHI measurements" Renewable Energy, v90, p. 520-531, 2016.

NREL | 8

Clearsky day filtering:

Package Features: System Characteristics

- Mounting configuration
 - Fixed-tilt or single-axis tracking
 - Uses daily power profile to classify time series stream

• Azimuth and tilt

- Estimate using AC power time series
- Work in progress: multiple methods in package are currently being validated





Algorithm Validation

- Continued validation of each algorithm
 - How well does each algorithm perform on labeled data sets?
 - Quantifiable metrics: accuracy and F1-score
 - Labeled data sets to encourage further development
- Technical documentation/publications benchmarking each algorithm's performance

https://datahub.duramat.org/project/example-data

Project	Example and Eva	luation Data - Datasets				
rview		lation Data - Datasets				
ets	Project ID	e46c953f-0d84-41c8-9526-2a219	50e6c92			
s						
(3)	Search datasets					
shift (2)						
raining (2)		Order by:	Relevance			
L (2)	Labeled Clearsky Period Data Set					
ng (2)	Resource					
-9 (/	1 Resource					
		I minute-interval data for the NREL Sanyo 1 system. The C	SV contains the			
series (2)	This dataset contains labeled clearsky periods for 1		SV contains the			
series (2) ing data (2) imated (1)	This dataset contains labeled clearsky periods for 1 following columns: measured_on (the timestamp		SV contains the			
series (2) ing data (2)	This dataset contains labeled clearaty periods for following columns: measured_on (the timestamp inverter Clipping ML Training Set - Real 64 Resources Real PV Field data for evaluating, validating, or train					
series (2) ing data (2) mated (1)	This dataset contains labeled clearsky periods for 1 following columns: measured_on (the timestamp	l Data				
series (2) ing data (2) imated (1) ning (1)	This dataset contains labeled clearaty periods for following columns: measured_on (the timestamp inverter Clipping ML Training Set - Real 64 Resources Real PV Field data for evaluating, validating, or train	l Data				
series (2) ing data (2) imated (1) ining (1) issky (1)	This dataset contains labeled clearaty periods for 1 following columns: measured_on (the timestamp	I Data	ents.			
series (2) ing data (2) mated (1) ning (1) sky (1) V More Tags	This dataset contains labeled clearaty periods for 1 following columns: measured_on (the timestamp	I Data ling machine learning algorithms to recognize clipping eve ss that are labeled for abrupt data shifts. The data, which i	ents.			
eries (2) g data (2) anated (1) ing (1) ky (1) More Taps titution	This dataset contains labeled clearaty periods for 1 following columns: measured_on (the timestamp	I Data ing machine learning algorithms to recognize clipping eve is that are labeled for abrupt data shifts. The data, which i	ents.			
eries (2) g data (2) sated (1) ng (1) More Tags More Tags (10)	This dataset contains labeled clearsky periods for 1 following columns: measured_on (the timestamp Inverter Clipping ML Training Set - Real 64 Resources Real FV Field data for evaluating, validating, or train https://doi.org/10.21948/1874779 Labeled Time Series Data Shifts 1 Resource This data set includes 101 daily summed time serie	I Data ing machine learning algorithms to recognize clipping eve is that are labeled for abrupt data shifts. The data, which i	ents.			

Publicly available, labeled data sets on the DuraMAT DataHub

Documentation: Example Gallery

- Example gallery for majority of the package functions (v0.1.2)
 - Example data for running each algorithm
 - Plots illustrating algorithm results

API Reference	Example G	allerv		Note:
Example Gallery Z-Score Outlier Detection Tukey Outlier Detection 	This gallery shows examp	e e	onality. Community contributions a	Click here to download the full example code
Hampel Outlier Detection Clear-Sky Detection Interpolated Data Periods Clearsky Limits for Daily Insolation Data Shift Detection & Filtering Clearsky Limits for Irradiance Data Stale Data Periods	welcome!	Tukey Outlier Detection	Hampel Outlier Detection	Clearsky Limits for Daily Insolation Checking the clearsky limits for daily insolation data. Identifying and filtering out invalid irradiance data is a useful way to reduce noise du ing analysis. In this example, we use pvanalytics.quality.irradiance.dsily_insolation_limits() to determine when th daily insolation lies between a minimum and a maximum value. Irradiance measure- ments and clear-sky irradiance on each day are integrated with the trapezoid rule to o culate daily insolation. For this example we will use data from the RMIS weather syst located on the NREL campus in Colorado, USA.
Clipping Detection Qcrad Limits for Irradiance Data Missing Data Periods Qcrad Consistency for Irradiance Data Day-Night Masking Release Notes Quick search	Clear-Sky Detection	Interpolated Data Periods	Clearsky Limits for Daily Insolation	<pre>import puanalytics from puanalytics.quality.irradiance import daily_insolation_limits import public import pathlib. First, read in data from the RMIS NREL system. This data set contains 5-minute right aligned data. It includes POA, GHL, DNL, DHL, and GNL measurements. puanalytics_dir = pathlib.Path(puanalyticsfile).parent rmis file = puanalytics dir / 'dradiance RMIS NREL.csv'</pre>
Go				<pre>data = dg.read_csx(rmis_file, index_col=0, parse_dates=True) # Noke the datetime index tz-aware. data.index = data.index.tz_localize("Etc/GMT+7") Now model clear-sky irradiance for the location and times of the measured data:</pre>
	Data Shift Detection & Filtering	Clearsky Limits for Irradiance Data	Stale Data Periods	

https://pvanalytics.readthedocs.io/en/stable/generated/gallery/index.html

Apply PVAnalytics to Your Own Data

How can you easily implement PVAnalytics functions to your own data?

CSV containing data streams (power, irradiance, temperature) is labeled as False. The data is sampled at 15-minute intervals.



Import CSV into our example documentation, and change any metadata parameters (lat-long coordinates, data frequency, etc)



https://pvanalytics.readthedocs.io/en/stable/generated/gallery/index.html

NREL | 12

Documentation: Function Descriptions

- Page for each model function containing:
 - Brief description
 - Input parameters: data type, description
 - Outputs: data type, description
 - Published reference for the function, if applicable
 - Additional notes as needed
 - Examples in the gallery using the function

nalytics.quali	<pre>ty.gaps.stale_values_diff(x, window=6, rtol=1e-05, 'toi!')</pre>
Identify stale val	
	length N, the last value (index N-1) is considered stale if all values e close to the first value (index 0).
Parameters rtol	and <i>atol</i> have the same meaning as in numpy.allclose() .
Parameters:	 x (Series) - data to be processed window (int, default 0) - number of consecutive values which, if unchanged, indicates state data rtol (float, default 1e-5) - relative tolerance for detecting a change in data values atol (float, default 1e-5) - absolute tolerance for detecting a change in data values mark (gtr, default 1e-1) - absolute tolerance for detecting a change is data values imark (gtr, default 1e-1) - absolute tolerance for detecting a change is idetected. Can one be of 'tail', end', or 'all'. if 'tail' (the default) then every point in the window except the first point is marked True. if 'all' then every point all subsequence sequence are marked True. if 'all' then every point in the window including the first point is marked True.
Returns: Return type: Raises:	True for each value that is part of a stale sequence of data Series ValueError – If window < 2 or mark is not one of 'tail', 'end', or 'all'.
Notes	
LICENSES/SOL distribution and	m/pvlib/pvanalytics/blob/master/LICENSES/SOLARFORECASTARBITER_LICENSE
xamples u /analytic	sing s.quality.gaps.stale_values_diff

Function description for

pvanalytics.quality.stale_values_diff

https://pvanalytics.readthedocs.io/en/stable/api.html

Automated Testing

- Comprehensive unit-testing for all package functions
 - ~100% test coverage
 - Uses Pytest and Coveralls
- Since package is in its infancy, no speed benchmarks have been taken (yet!)

30 checks passed Image: Constraint of the state s

Package checks required to pass before merging PR

Int and test passing coverage 100% DOI 10.5281/zenodo.6110569

Current test coverage

Community growth

- Github
 - 88 completed pull requests
 - Code contributions from 6 people (see lower right)
- Lots of opportunity to increase community growth as PVAnalytics is still in its infancy
- You can contribute!
 - Generate issues for features you'd like to see, add code via our PR process, etc.



Github stars over time



Special thanks to all our contributors!

PVAnalytics v0.1.3 and Beyond

- No expected ETA for next release but we're actively working on new functions/documentation
- Future version features:
 - Daylight savings time (DST) and time-drift detection algorithms for time series
 - Adding plotting module to easily validate time-series data visually

Thank you!

www.nrel.gov

kirsten.perry@nrel.gov

This work was authored in part by Alliance for Sustainable Energy, LLC, the manager and operator of the National Renewable Energy Laboratory for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under Solar Energy Technologies Office (SETO) Agreement Number 38258. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

