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

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

PVAnalytics Background (Continued)

• **Design Principles behind PVAnalytics:**
  - Open-source: tested, documented, and re-usable
  - 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]

Package Features: Basic Time Series Filtering

**Outlier detection and filtering:** Hampel, Z-score, 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

**Data shift detection and filtering**: Uses changepoint detection to find massive, abrupt capacity changes. Described further in [1]

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]


Package Features: Irradiance Checks

**Irradiance quality checks:** consistency and physical limits of GHI, DNI, and DHI using QCrad criteria

**Clearsky period filtering:** Reno clearsky method (1)

**Clearsky day filtering:** Compare GHI sensor-based data to clearsky data. Filter where GHI is within daily insolation limit

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

Daily power profile of a single-axis tracking system
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

Publicly available, labeled data sets on the DuraMAT DataHub
• Example gallery for majority of the package functions (v0.1.2)
  • Example data for running each algorithm
  • Plots illustrating algorithm results

Apply PVAnalytics to Your Own Data

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

CSV containing data streams (power, irradiance, temperature)

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

Run associated example!

Analyze outputs

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

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!)
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 graph]

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
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.