pvOps: a Python package for empirical analysis of photovoltaic field data

Kirk L. Bonney[1¶], **Thushara Gunda**[1], **Michael W. Hopwood**[2], **Hector Mendoza**[1], **and Nicole D. Jackson**[1]

1 Sandia National Laboratories, USA **2** University of Central Florida, USA ¶ Corresponding author

DOI: 10.21105/joss.05755

Software

- Review
- Repository
- Archive

Editor: Rachel Kurchin

Reviewers:

- @FlorianK13
- @AdamRJensen

Submitted: 19 May 2023
Published: 16 November 2023

Summary

The purpose of pvOps is to support empirical evaluations of data collected in the field related to the operations and maintenance (O&M) of photovoltaic (PV) power plants. pvOps presently contains modules that address the diversity of field data, including text-based maintenance logs, current-voltage (IV) curves, and timeseries of production information. The package functions leverage machine learning, visualization, and other techniques to enable cleaning, processing, and fusion of these datasets. These capabilities are intended to facilitate easier evaluation of field patterns and extraction of relevant insights to support reliability-related decision-making for PV sites. The open-source code, examples, and instructions for installing the package through PyPI can be accessed through the GitHub repository.

Statement of Need

Continued interest in PV deployment across the world has resulted in increased awareness of needs associated with managing reliability and performance of these systems during operation. Current open-source packages for PV analysis focus on theoretical evaluations of solar power simulations (e.g., `pvlib` (Holmgren et al., 2018)), data cleaning and feature development for production data (e.g. `pvanalytics` (Perry et al., 2022)), specific use cases of empirical evaluations (e.g., `RdTools` (Deceglie et al., 2018) and `Pecos` (Klise & Stein, 2016) for degradation analysis), or analysis of electroluminescene images (e.g., `PVimage` (Pierce et al., 2020)); see openpvtools for a list of additional open source PV packages. However, a general package that can support data-driven, exploratory evaluations of diverse field collected information is currently lacking. For example, a maintenance log that describes an inverter failure may be temporally correlated to a dip in production levels. Identifying such relationships across different types of field data can improve understanding of the impacts of certain types of failures on a PV plant. To address this gap, we present `pvOps`, an open-source Python package that can be used by researchers and industry analysts alike to evaluate and extract insights from different types of data routinely collected during PV field operations.

PV data collected in the field varies greatly in structure (e.g., timeseries and text records) and quality (e.g., completeness and consistency). The data available for analysis is frequently semi-structured. Furthermore, the level of detail collected between different owners/operators might vary. For example, some may capture a general start and end time for an associated event whereas others might include additional time details for different resolution activities. This diversity in data types and structures often leads to data being under-utilized due to the amount of manual processing required. To address these issues, `pvOps` provides a suite of data processing, cleaning, and visualization methods to leverage insights across a broad range of data types, including operations and maintenance records, production timeseries, and IV curves. The functions within `pvOps` enable users to better parse available data to understand patterns in outages and production losses.

Package Overview

The following table summarizes the four modules within `pvOps` by presenting: the type of data they analyze, example data features, and highlights of relevant functions.

Table 1. Summary of modules and functions within ‘pvOps‘

Module	Type of data	Example data features	Highlights of functions
text	O&M records	*timestamps*, *issue description*, *issue classification*	fill data gaps in dates and categorical records, visualize word clusters and patterns over time
timeseries	Production data	*site*, *timestamp*, *power production*, *irradiance*	estimate expected energy with multiple models, evaluate inverter clipping
text2time	O&M records and production data	see entries for `text` and `timeseries` modules above	analyze overlaps between O&M and production (timeseries) records, visualize overlaps between O&M records and production data
iv	IV records	*current*, *voltage*, *irradiance*, *temperature*	simulate IV curves with physical faults, extract diode parameters from IV curves, classify faults using IV curves

The functions within each module can be used to build pipelines that integrate relevant data processing, fusion, and visualization capabilities to support user endgoals. For example, a user with IV curve data could build a pipeline that leverages functions within the `iv` module to process and extract diode parameters within IV curves as well as train models to support classifications based on fault type. A pipeline could be also be built that leverages functions across modules if a user has access to multiple types of data (e.g., both O&M and production records). A sample end-to-end workflow using `pvOps` modules could be:

1. Use functions within the `text` module to systematically review data quality issues within O&M records, train a machine learning model on available records, and use the model to estimate possible labels for missing entries
2. Leverage the functions within the `timeseries` module, use machine learning to develop their own expected energy models for a given time series of irradiance and system size details, or use a pre-trained expected energy model (Hopwood & Gunda, 2022) or leverage industry standard equations as a basis for evaluating possible production losses
3. Couple outputs from the above two analyses (using functions in the `text2time` module) based on timestamps to develop summaries and visualizations of production impacts observed during these periods

The package documentation for `pvOps` provides thorough examples exploring the various capabilities of each module. Additional details about the `iv` module capabilities are captured in (Hopwood et al., 2020; Hopwood, Stein, et al., 2022) while more information about the design and development of the `text`, `timeseries`, and `text2time` modules are captured in (Mendoza et al., 2021). Key package dependencies of `pvOps` include pandas (The pandas development team, 2020), `sklearn` (Pedregosa et al., 2011), `nltk` (Bird et al., 2009), and `keras` (Chollet & others, 2015) for analysis and `matplotlib` (Hunter, 2007), `seaborn` (Waskom, 2021), and `plotly` (Plotly Technologies Inc., 2015) for visualization.

Ongoing Development

The `pvOps` functionality and documentation continues to be improved and updated as new empirical techniques are identified. For example, research efforts have demonstrated utility of natural language processing techniques (e.g., topic modeling) and survival analyses to support evaluation of patterns in O&M records (Gunda et al., 2020). Additional statistical methods, such as Hidden Markov Modeling, have also been successfully used to support classification of failures within production data (Hopwood, Patel, et al., 2022). These and other capabilities will continue to be added to the package to improve its utility for supporting empirical analyses of field data.

CRediT Authorship Statement

KLB: Writing - Original Draft, Software - Software Development, Software - Testing; TG: Conceptualization, Writing - Original Draft, Software - Design; MWH: Writing - Review & Editing, Software - Software Development; HM: Writing - Review & Editing, Software - Software Development; NDJ: Conceptualization, Funding Acquisition, Project Administration, Supervision, Writing - Review & Editing.

Acknowledgements

This material is supported by the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy - Solar Energy Technologies Office. Sandia National Laboratories, a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

References

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python*. O'Reilly Media.

Chollet, F., & others. (2015). *Keras*. https://keras.io.

Deceglie, M. G., Jordan, D., Nag, A., Deline, C. A., & Shinn, A. (2018). *RdTools: An open source python library for PV degradation analysis*. National Renewable Energy Lab.(NREL), Golden, CO (United States).

Gunda, T., Hackett, S., Kraus, L., Downs, C., Jones, R., McNalley, C., Bolen, M., & Walker, A. (2020). A machine learning evaluation of maintenance records for common failure modes in PV inverters. *IEEE Access*, *8*, 211610–211620. https://doi.org/10.1109/ACCESS.2020.3039182

Holmgren, W. F., Hansen, C. W., & Mikofski, M. A. (2018). Pvlib python: A python package for modeling solar energy systems. *Journal of Open Source Software*, *3*(29), 884. https://doi.org/10.21105/joss.00884

Hopwood, M. W., & Gunda, T. (2022). Generation of data-driven expected energy models for photovoltaic systems. *Applied Sciences*, *12*(4), 1872. https://doi.org/10.3390/app12041872

Hopwood, M. W., Gunda, T., Seigneur, H., & Walters, J. (2020). Neural network-based classification of string-level IV curves from physically-induced failures of photovoltaic modules. *IEEE Access*, *8*, 161480–161487. https://doi.org/10.1109/ACCESS.2020.3021577

Hopwood, M. W., Patel, L., & Gunda, T. (2022). Classification of photovoltaic failures with hidden markov modeling, an unsupervised statistical approach. *Energies*, *15*(14), 5104. https://doi.org/10.3390/en15145104

Hopwood, M. W., Stein, J. S., Braid, J. L., & Seigneur, H. P. (2022). Physics-based method for generating fully synthetic IV curve training datasets for machine learning classification of PV failures. *Energies*, *15*(14), 5085. https://doi.org/10.3390/en15145085

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Klise, K. A., & Stein, J. S. (2016). *Performance monitoring using pecos (v. 0.1)*. Sandia National Laboratories. https://doi.org/10.2172/1734479

Mendoza, H., Hopwood, M., & Gunda, T. (2021). pvOps: Improving operational assessments through data fusion. *2021 IEEE 48th Photovoltaic Specialists Conference (PVSC)*, 0112–0119. https://doi.org/10.1109/PVSC43889.2021.9518439

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Perry, K., Vining, W., Anderson, K., Muller, M., & Hansen, C. (2022). *PVAnalytics: A python package for automated processing of solar time series data*. National Renewable Energy Lab.(NREL), Golden, CO (United States).

Pierce, B. G., Karimi, A. M., Liu, J., French, R. H., & Braid, J. L. (2020). Identifying degradation modes of photovoltaic modules using unsupervised machine learning on electroluminescense images. *2020 47th IEEE Photovoltaic Specialists Conference (PVSC)*, 1850–1855. https://doi.org/10.1109/PVSC45281.2020.9301021

Plotly Technologies Inc. (2015). *Collaborative data science*. Plotly Technologies Inc. https://plot.ly

The pandas development team. (2020). *Pandas-dev/pandas: pandas* (latest). Zenodo. https://doi.org/10.5281/zenodo.3509134

Waskom, M. L. (2021). Seaborn: Statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. https://doi.org/10.21105/joss.03021