Measuring degradation of fielded systems on an ongoing basis with RdTools

Adam Shinn, Ben Browne
kWh Analytics, Inc.
PV Reliability Workshop 2019

Motivation

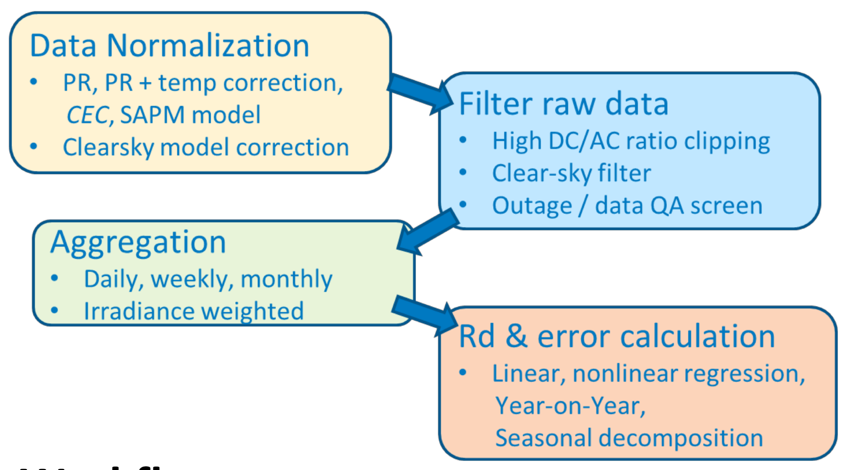
The solar investment community needs up-to-date and data-derived system degradation rates to use in financial models that calculate the risk and expectation of energy revenue. To date, the majority of financial models choose degradation rates from past studies that are limited and may not be representative of future solar assets (systems) they are financing. As the solar industry expands in all sectors (residential, commercial, and utility), there is a growing amount of energy production available to analyze. With the help of the open source degradation analysis code (RdTools) it is now possible to derive degradation on an ongoing basis and continuously provide up-to-date durability statistics to the investment community. Ongoing and accurate statistics of fielded photovoltaic systems allows financial stakeholders to constrain energy revenue projections, lower financial risk, and increase the bankability of solar.

Abstract

This poster will present considerations and software engineering best practices to encourage data owners to measure degradation of their own photovoltaic assets on an ongoing basis. This poster will also preview an upcoming DOE-funded kWh Analytics degradation project which involves applying RdTools analysis on kWh Analytics' industry database, aggregating results across variables such as weather, hardware, system configurations etc., and sharing the aggregated results with the broader community.

RdTools Open Source Degradation Analysis

RdTools is an open source software project hosted on Github, written in Python, and leverages PVLIB-Python and SciPy packages. RdTools degradation analysis is ideal when system details are well known and there are a few years of sub-daily operational history is available. The RdTools approach uses 4 steps to calculate degradation, and is designed to be robust to sensor drift and anomalous observations. [1][2][3]



Example Workflow

1. Production data are normalized to a temperature-corrected performance index
2. Removal of unphysical time periods (such as inverter clipping, non clear sky, or outages)
3. Irradiance-weighted aggregation to daily frequency
4. Year-on-Year comparison of daily values

NREL https://github.com/NREL/rdtools

kWh Analytics DOE-funded Degradation Project

Timeline
- 2-year project, officially starting March 1st, 2019

Project Goal
- Analyze degradation on an industry data set on an ongoing basis, allowing financial stakeholders to have access to up-to-date degradation statistics that can better inform financial models for photovoltaic assets

Key Deliverables
- Researcher and stakeholder engagement
- Contributions to RdTools (focusing on tools that help RdTools analysis pipeline)
- Dissemination of aggregated results, showing degradation as a function of predictor variables, and their statistical significance
- Publish statistical models as open source software

Research Scope
- Calculate degradation for as many systems as possible in the kWh Analytics database:
  - Over 200k systems across the U.S.
  - Residential, C&I, Utility
- Using system design and equipment information, collect any additional metadata that may be predictive of module or system degradation

System and module predictor variables to focus on:
- System design (mount type, inverter type, etc.)
- Climate Characteristics
- Module bill of materials (cell type, vintage, max voltage, etc.)
- **Currently soliciting feedback!**

- Use advanced statistical techniques to search for meaningful correlation between observed degradation and system design, equipment, and environmental factors

Lessons Learned from Running Analyses on an Industry Database

Is Your Analysis Pipeline Scalable?

A scalable analysis pipeline can better support additional systems and new production data as it becomes available.

Consistent Data Formats
Whether you are using a database or working with flat files, standardizing input data is an important first step to scaling a pipeline.

Database Considerations
Prioritize infrastructure. Store data with hardware or a web service that can grow with your data. This is especially important when working with high frequency (<=1hr) time series production data.

Efficient Computation
Even small improvements in run time for a single iteration can translate to large improvements for the pipeline. Monitor your bottlenecks via program and network profiling.

Is Your Code Extensible?

An extensible pipeline can easily incorporate improvements and allow for reproducible results.

Follow Software Engineering Best Practices[4]
- Modular programing
- Unit tests
- Regression tests
- Code review by peers
- Documentation embedded in all functions

Version Control Everything
Not only version control the analysis code, (e.g. GIT & SVN) but also the environment that it runs in (e.g. pip requirements files & Docker containers)

Exception Handling
Catch bugs early and often by handling exceptions with helpful messages describing how failures occurred (e.g. raise exceptions if input values are unphysical)

References

[1] D. Jordan, C. Deline, S. Kurtz, G. Kimball, M. Anderson, "Robust PV Degradation Methodology and Application", IEEE Journal of Photovoltaics, 2017
[2] Michael Deceglie, et al., "Fleet-Scale Photovoltaic Energy-Yield Degradation Analysis Applied to Hundreds of Residential and Non-Residential PV Systems", IEEE Journal of Photovoltaics, 2019
[3] Dirk C. Jordan, et al., "PV Degradation Rate Interlaboratory Study", IEEE Journal of Photovoltaics, 2019
[4] Andrew Hunt and David Thomas, *The Pragmatic Programmer*, Addison-Wesley, 2000

- *Acknowledgment:* “This material is based upon work supported by the U.S. Department of Energy’s Office of Energy Efficiency and Renewable Energy (EERE) under the Solar Energy Technology Office (SETO), Award Number DE-EE0008555.”