

OpenOA: An Open-Source Codebase For Operational Analysis of Wind Farms

Jordan Perr-Sauer¹, Mike Optis¹, Jason M. Fields¹, Nicola Bodini¹, Joseph C.Y. Lee¹, Austin Todd¹, Eric Simley¹, Robert Hammond¹, Caleb Phillips¹, Monte Lunacek¹, Travis Kemper¹, Lindy Williams¹, Anna Craig¹, Nathan Agarwal¹, Shawn Sheng¹, and John Meissner¹

¹ National Renewable Energy Laboratory, Golden, CO, USA

DOI: [10.21105/joss.02171](https://doi.org/10.21105/joss.02171)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Stefan Pfenninger](#) ↗

Reviewers:

- [@gschivley](#)
- [@brynpickering](#)

Submitted: 19 February 2020

Published: 26 February 2021

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

OpenOA is an open source framework for operational data analysis of wind energy plants, implemented in the Python programming language ([Van Rossum & Drake, 2009](#)). OpenOA provides a common data model, high level analysis workflows, and low-level convenience functions that engineers, analysts, and researchers in the wind energy industry can use to facilitate analytics workflows on operational data sets. OpenOA contains documentation, worked out examples in Jupyter notebooks, and a corresponding example dataset from the [Engie Renewable's La Haute Borne Dataset](#).

Originally released to the public in 2018 ([Perr-Sauer et al., 2018](#)), OpenOA is now actively developed through a public Github repository. With over 50 stars on Github, a dozen contributors, and an active issues forum, OpenOA is becoming a mature project that provides a high level interface to solve practical problems in the wind energy industry. OpenOA V2 is released as a Python package and is freely available under a business-friendly, open-source, BSD license. By committing to open source development, OpenOA hopes to facilitate reproducibility of research in this field, provide benchmark implementations of commonly performed data transformation and analysis tasks, and to serve as a conduit that delivers state-of-the-art analysis methods from researchers to practitioners.

Most users will interface with OpenOA through its analysis methods module. This includes Python classes which conform to a common interface (e.g., they implement `__init__`, `prepare`, and `run` methods). Version 2 of OpenOA implements three high level analysis methods for the calculation of: (1) Long term corrected annual energy production (AEP), (2) electrical losses, and (3) turbine level losses. Uncertainty quantification is achieved in each analysis using a Monte Carlo approach. A more detailed description of these analyses are provided [in the documentation](#). Low level functions that operate on Pandas series objects are organized in the `toolkit` module. These Python functions are written to be as generic as possible, and can be applied across multiple domains.

The OpenOA data model is implemented in the `types` module using a class called `PlantData`, which contains at least one Pandas data frame ([McKinney & others, 2010](#)). These classes add convenience functions and a domain-specific schema based on the IEC 6400-25 standard ([Wind Turbines – Part 25-2, 2015](#)). OpenOA is part of the ENTR alliance consortium, which envisions a complete software stack centered around an open source implementation of this standard.

OpenOA depends on scikit-learn ([Pedregosa et al., 2011](#)) and numpy ([Oliphant, 2006](#)), with graphing functions implemented using matplotlib ([Hunter, 2007](#)). The OpenOA development team strives to use modern software development practices. Documentation is compiled

from the source code and automatically published to [ReadTheDocs](#). We use Github actions to implement our continuous integration pipeline, including automated unit and regression tests, test coverage reporting via CodeCov, automated packaging and publication to the PyPI package index. We utilize a modified git-flow development workflow, with pull requests and issue tracking on Github driving the development.

Statement of Need

OpenOA was created and is primarily developed by researchers at the National Renewable Energy Laboratory (NREL) through the Performance, Risk, Uncertainty, and Finance (PRUF) project. The PRUF team recognized the need to compute a long term corrected AEP (comparable to a 20-year estimate) from operational data as part of an industry-wide benchmarking study ([Lunacek et al., 2018](#)). Due to access restrictions on the input data, open source publication of the code was necessary to foster trust in the results of the benchmarking study. Furthermore, after talking with our industry partners, it became clear that there was no industry standard method for computing a long term corrected AEP. Currently, participants in the wind energy industry who wish to compute metrics like AEP must rely on commercial secondary supervisory control and data acquisition (SCADA) software providers, or must develop their own methodologies internally. We know of no other open source software package to compute long term corrected AEP.

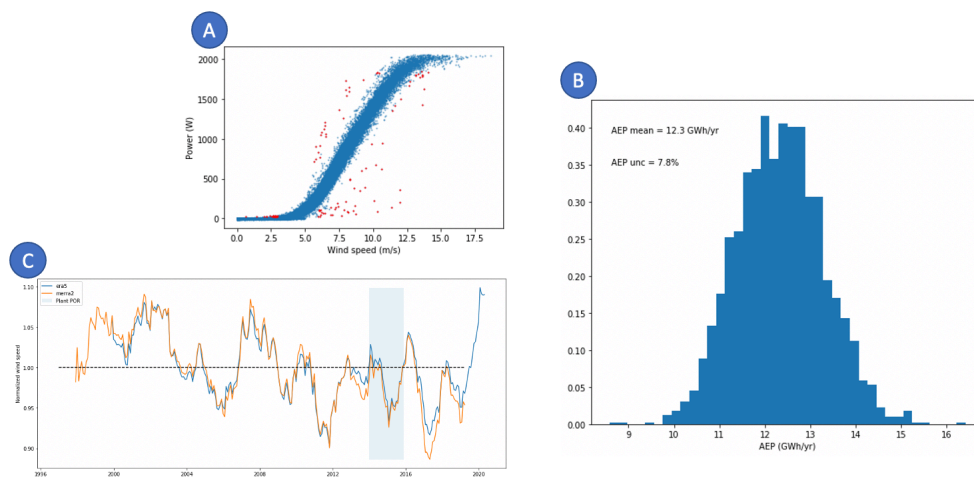


Figure 1: A subset of graphical outputs from the OpenOA documentation. Clockwise from the top, (A) power curve with extreme values highlighted in red, (B) distribution of long term corrected AEP, (C) time-series of wind speed from multiple reanalysis products showing anomalously low wind speed for a highlighted period of record.

Operational analysis involves obtaining time-series data from an industrial plant's SCADA system, performing quality control processes on these data, and computing metrics that provide insight into the performance characteristics of a wind plant. Figure 1 contains some graphical outputs that are generated by OpenOA. Since its inception, OpenOA has been used in several published studies at NREL. An early version of the code was used in [Craig et al. \(2018\)](#) to quantify the uncertainty in AEP estimates resulting from analyst choices. In [Bodini & Optis \(2020\)](#), it is used to calculate long-term operational AEP from over 470 wind plants in the US to assess correlation between uncertainty components. OpenOA will also be used in an upcoming technical report for the PRUF project's industry-wide benchmarking study.

Acknowledgements

The authors would like to acknowledge that Jordan Perr-Sauer and Mike Optis have made an equal contribution to this work. This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the U.S. Department of Energy Office of Energy Efficiency and Renewable Energy Wind Energy Technologies Office, within the Atmosphere to Electrons research program. 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.

References

- Bodini, N., & Optis, M. (2020). Are uncertainty categories in a wind farm annual energy production estimate actually uncorrelated? *Wind Energ. Sci.*, 5, 1435–1488. <https://doi.org/10.5194/wes-5-1435-2020>
- Craig, A., Optis, M., Fields, M. J., & Moriarty, P. (2018). Uncertainty quantification in the analyses of operational wind power plant performance. *Journal of Physics: Conference Series*, 1037, 052021. <https://doi.org/10.1088/1742-6596/1037/5/052021>
- 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>
- Lunacek, M., Fields, M. J., Craig, A., Lee, J. C. Y., Meissner, J., Philips, C., Sheng, S., & King, R. (2018). Understanding biases in pre-construction estimates. *Journal of Physics: Conference Series*, 1037(6), 062009. <https://doi.org/10.1088/1742-6596/1037/6/062009>
- McKinney, W., & others. (2010). Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference*, 445, 51–56. <https://doi.org/10.25080/Majora-92bf1922-00a>
- Oliphant, T. E. (2006). *A guide to NumPy* (Vol. 1). Trelgol Publishing USA.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., & others. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct), 2825–2830.
- Perr-Sauer, J., Fields, M., Craig, A., Optis, M., Kemper, T., Sheng, S., Meissner, J., & Phillips, C. (2018). *Open OA, FKA: Wind plant performance project (WP3) benchmarking*. USDOE Office of Energy Efficiency; Renewable Energy. <https://doi.org/10.11578/dc.20181023.1>
- Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. CreateSpace. ISBN: 1441412697
- Wind turbines – part 25-2: Communications for monitoring and control of wind power plants – information models*. (2015). [Standard]. International Electrotechnical Commission.