Wind Energ. Sci. Discuss., https://doi.org/10.5194/wes-2019-12
Manuscript under review for journal Wind Energ. Sci.
Discussion started: 2 April 2019

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

OpenOA: An Open-Source Code Base for Operational Analysis of Wind Power Plants

Mike Optis, Jordan Perr-Sauer, Caleb Philips, Anna E. Craig, Joseph C.Y. Lee, Travis Kemper, Shuangwen Sheng, Eric Simley, Lindy Williams, Monte Lunacek, John Meissner, and M. Jason Fields
15013 Denver West Pkwy, Golden, CO, USA, 80401
National Renewable Energy Laboratory
Correspondence: Mike Optis (mike.optis@nrel.gov)

Abstract. As global wind capacity continues to grow, the need for accurate operational analyses of a rapidly growing fleet of wind power plants has increased in proportion. The wind energy industry at present, however, is not ideally positioned to address this need. First, there is a lack of best practices and limited published standards for performing operational analyses. Second, operational data and methods are typically proprietary and not shared among the wind energy community. Consequently, there is considerable duplication of effort in developing methods as well as uncertainty in the calculated metrics. To address these problems, the National Renewable Energy Laboratory has publicly released OpenOA, an open-source code base for performing operational analyses on wind plant data. The intent of OpenOA is to provide a framework in which best practices can be developed, refined, and disseminated. Ultimately, such collaboration is expected to lead to a working example (i.e. reference implementation) of methods from which a published standard may develop. This article provides an overview of OpenOA, highlighting its release as a public repository, modular software architecture, current functionality, and planned functionality in subsequent releases. It is our goal for OpenOA to evolve into an indispensable tool for performing operational analyses that is used and supported by a large community of wind energy experts.

1 Introduction

1.1 A lack of best practices

Operational analyses use collected data from wind power plants to perform assessments ranging from the diagnosis of faults and underperformance, benchmarking of performance improvements (e.g., wind sector management, vortex generators), long-term estimates of annual energy production (AEP), and building/tuning statistical or physics-based models for various applications (e.g., wake model validation, wind power forecasting). The needs for such analyses is increasing. Global wind capacity increased to 539 GW in 2017, representing 11% and 91% increases over 1-year and 5-year periods, respectively. Capacity is expected to increase another 56% to 841 GW by 2022 Council (2018). This presents a significant data and analysis challenge for owners who must manage increasingly large fleets of wind power plants as well as the consultants and researchers who perform these analyses.

Wind Energ. Sci. Discuss., https://doi.org/10.5194/wes-2019-12
Manuscript under review for journal Wind Energ. Sci.
Discussion started: 2 April 2019

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

The wind energy industry is highly competitive and, consequently, data and methods have tended to be proprietary and not widely shared Moseson (2014); Kusiak (2016). This situation has consequences in terms of efficiency and uncertainty. Many steps in a wind energy analysis (both preconstruction and operational) could be standard and uncontroversial, such as data quality control and preprocessing and data flagging and filtering. When methods are not shared, there is significant duplication of effort as different stakeholders (e.g., owner/operators, consultants, and original equipment manufacturers) are all developing independent methods to perform the same tasks. Beyond inefficiency, this duplication of effort introduces added uncertainty in the calculated metrics because different analysts may process wind plant data and calculate metrics in different ways. This uncertainty is best exemplified by the current state of preconstruction energy yield assessments for wind power plants. In Lunacek et al. (2018), it was found that energy yield assessments conducted since 2011 have, on average, overpredicted energy production by about 3%–5%, with large biases of 10%–15% not uncommon. For the same wind plant, considerable differences were found across EYAs performed by different consultants Laboratory (in draft). A framework which would allow for data sharing and collaborative methods development in the wind industry would therefore represent a significant opportunity to reduce analysis uncertainty in the wind industry.

Similar to energy yield assessments, operational analyses are also subject to variability depending on the analyst conducting the assessment and the quality of operational data provided. Numerous steps in an OA can be influenced by analyst subjectivity, such as initial quality control of the data, flagging and removal of unrepresentative or outlier data points, the fitting of data to a particular model (e.g., turbine power curve), filling in missing or flagged data, and methods and data sources used for a long-term correction. For example, Craig et al. (2018) showed that different combinations of data flagging algorithms applied to nacelle wind speed and turbine power data, along with different choices for power curve models, resulted in 3.0% total spread and 0.7% interquartile range when estimating gross energy production for a wind plant over a year-long period Craig et al. (2018).

1.2 The value of published standards

Published standards can be useful for reducing metric variability between analysts and building consensus on industry best practices. For preconstruction energy yield assessments, the International Energy Commission (IEC) 61400-15 standard IEC 61400-15:draft is currently being drafted and will help ensure different industry consultants are following similar best practices when performing energy yield assessments. For operational assessments, there are only limited standards covering specific applications: IEC 61400-12 IEC 61400-12-1:2017 addresses turbine power curve testing and IEC 61400-26 IEC 61400-26-3:2016 addresses the derivation and categorization of availability loss metrics. Notably lacking standards are AEP estimates, reliability and performance metrics, and fault and underperformance diagnosis. In fact, very little documentation of OA best practices exists beyond these standards, and seems to be limited to a consultant report Lindvall et al. (2016), an academic thesis Khatab (2017), and some conference proceedings Lunacek et al. (2018).

Establishing industry best practices through a standard can be a long process. For example, IEC 61400-15 was first proposed in early 2013, a kick-off meeting was held in February 2014, and the planned release for the standard is fall 2019. This 6-year

Wind Energ. Sci. Discuss., https://doi.org/10.5194/wes-2019-12
Manuscript under review for journal Wind Energ. Sci.
Discussion started: 2 April 2019

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

schedule reflects the challenge of building industry consensus for a detailed and complex process, as well as limited time and resources that industry members have to contribute.

Once a standard is established, it may not always be effective, especially as many of the existing standards do not have viable working examples (i.e. reference implementations). For example, a comparative analysis in calculating rotor-equivalent wind speeds for different turbines revealed differences between 1.2%–1.8% across eight organizations in five different countries Wagner et al. (2014). These differences are observed despite rotor-equivalent wind speed guidance provided in IEC 61400-12. Furthermore, IEC 61400-25, which in part provides guidance on standardized taxonomy for data outputs from a wind plant, has largely been unadopted across the wind industry. The absence of such standardization was highlighted as the key source of operational analysis inefficiency at the International Energy Administration Topical Expert Meeting 92.

1.3 Open-source methods on route to a standard

The challenge in developing methods that adhere to a published standard and are reproducible across different organizations could be overcome if the standard was preceded by collaborative methods development within an open-source framework. Open-source software, used today by many organizations, is characterized by permissive licensing and the distribution of both the software and underlying code. This combination of permissive licensing and freely available source code enables users to download, modify, and redistribute the software with few restrictions. Open-source software offers many benefits to the user, including transparency in methods, more efficiency in methods development, low cost of development, quick error fixes, and community support.

Examples of the most commonly used open-source software include the Android operating system, Google Chrome (through the Chromium project), the Apache HTTP server, and the Linux operating systems (which power most web servers globally), as well as a suite of engineering and analysis tools in the scientific community. Examples of scientific tools in Python include SciPy and NumPy, as well as machine-learning projects, such as TensorFlow and ScikitLearn. Such software is free to use, modify, and redistribute, enabling tremendous innovation and many of the software startups that rely on these tools.

The development of operational analysis best practices for wind energy lends itself particularly well to an open-source approach. Relative to energy yield assessments, operational analysis methods are newer and the best practices are much less established. Furthermore, the wind energy community is increasingly adopting open-source code bases in languages such as Python and R, as evidenced by workshops and presentations recently held at the American Wind Energy Association Wind Resource and Project Energy Assessment Conference 2018. Provided industry willingness to share methods within a collaborative, transparent framework, an open-source repository of operational analysis methods could become a reference implementation from which a published standard might ultimately and efficiently follow.

1.4 The development of OpenOA

The development of OpenOA started internally at the National Renewable Energy Laboratory (NREL) and was initiated to support the lab's efforts in the Wind Plant Performance and Prediction (WP3) Benchmark, which is a key risk reduction

Wind Energ. Sci. Discuss., https://doi.org/10.5194/wes-2019-12
Manuscript under review for journal Wind Energ. Sci.
Discussion started: 2 April 2019

activity of the Performance, Risk, Uncertainty, and Finance project under the Atmosphere to Electrons initiative[1]). Further funding was provided through the NREL Research Data Initiative. The goal of WP3 is to provide an independent benchmark of preconstruction EYA bias (discussed in the preceding section) and to understand the sources of bias and uncertainty. OpenOA was originally scoped to calculate the operational AEP of case study wind power plants, to which EYA estimates of those plants provided by consultants could be compared. OpenOA is now expected to rapidly expand its scope to support additional types of analyses including turbine performance and reliability.

The OpenOA code base was designed from the start to have broad industry support and engagement. Specifically, several meetings were held in early 2018 in which the underlying methods and analyses in OpenOA were presented and discussed with most major stakeholders in the United States. Methods were continually refined over the course of 2018 as a result of these industry engagements. From these collaborative meetings, a clear desire emerged for this OA code base to be made open source and built in such a way to foster collaboration within industry.

With industry backing and enthusiasm, NREL published OpenOA as an open-source software package in September 2018. The first release of the code was fairly narrow in scope, consisting mostly of lower-level functions to perform general operations, such as filtering and filling in gaps and more specific functions like meteorological data analysis and power curve fitting. The software also includes an industry standard method to assess the AEP of a wind plant using operational data.

1.5 Overview of article

The purpose of this article is to highlight the key characteristics of OpenOA and to outline future development plans. Section 2 describes the location of the repository, the principles involved in its construction, and its architecture. Section 3 outlines the current functionality of Version 1.0. Section 4 provides the development plan for Version 2.0. Concluding remarks are provided in Section 5.

2 Design of OpenOA

This section outlines the software architecture and its development as well as the software repository on GitHub.

2.1 Software architecture

OpenOA is an open-source software package written in Python and licensed under a permissive open-source software license. We take advantage of the Python standard library's *unittest* framework for testing, *setuptools* for package installation, and rely on software such as ScikitLearn, NumPy, and Pandas for organizing and manipulating data and performing analyses. The code base features a modular design in which different combinations of modules can be employed to perform different types of analyses.

An overview of the Python modules contained in OpenOA is provided in Figure 1. This architecture was designed with extensibility in mind. We want to support the loading of operational data from various formats and locations with varying

[1]https://a2e.energy.gov

Wind Energ. Sci. Discuss., https://doi.org/10.5194/wes-2019-12
Manuscript under review for journal Wind Energ. Sci.
Discussion started: 2 April 2019

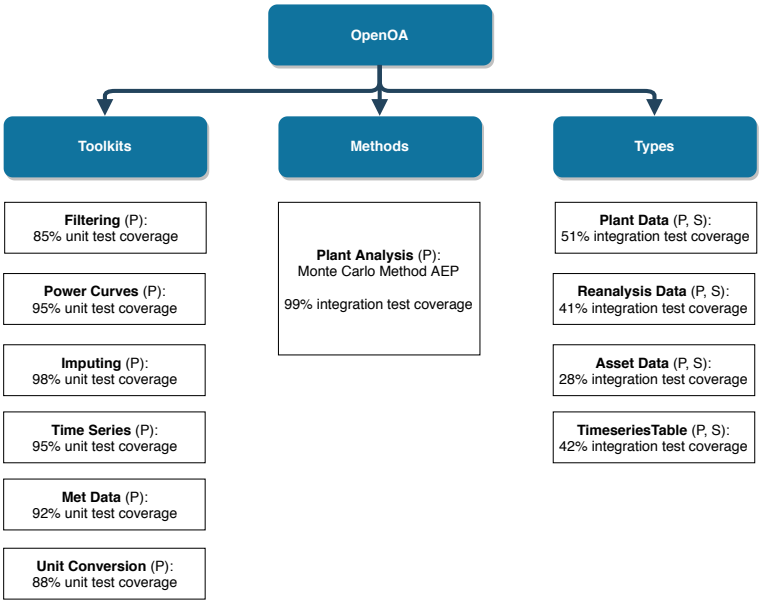

Figure 1. Software components organized by module for OpenOA version 1.0. Support for Pandas (P) and Spark (S) back end is noted in parentheses. Unit or integration test coverage is noted for each component.

levels of data quality, and the processing and analysis of the data using different methodologies. To this end, we implement an object hierarchy using a module called Types. This module contains Python classes that help represent potentially large operational data from wind plants in a framework that can be easily managed, processed, and analyzed. The core of this module is the Plant Data class, which acts as a container for the many different data streams from a wind plant. The remaining modules (Reanalysis Data, Asset Data, and Timeseries Table) are used to support particular data structures in the Plant Data module. Data attributes within the Plant Data module include:

- **Supervisory control and data acquisition (SCADA)** data for wind turbines, such as power output, wind speed, wind direction, and blade pitch
- **Revenue meter** data accounting for the power output from the plant
- **Meteorological (met) tower** data, such as wind speed, wind direction, temperature, and pressure
- **Status code and event log** data, which provide detailed performance and event reporting for turbines
- **Asset** data, which include information about each turbine and met tower in the wind plant
- **Reanalysis** data, which provide long-term, coarse-resolution modeled meteorological data for the geographical region of the wind plant.

Wind Energ. Sci. Discuss., https://doi.org/10.5194/wes-2019-12
Manuscript under review for journal Wind Energ. Sci.
Discussion started: 2 April 2019

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

Once imported into OpenOA, the data described earlier all have standard schema (e.g., column names), allowing for unified representation of data from different wind plants and streamlined processing and analysis. Data from different wind plants can vary considerably in format and quality; therefore, unique project import scripts making use of a *prepare* function in Plant Data is used to import wind plant data into the standardized taxonomy of the Plant Data object.

In addition to Types, there are two other top-level modules: Toolkits and Methods. These modules provide the functions to perform analyses on the Plant Data attributes. The Toolkits module provides groups of functions that perform general operations at the data frame level (e.g., data filter, wind direction calculation). Classes in the Methods module operate at the Plant Data level, providing higher-level, more detailed analysis procedures that compute metrics of scientific interest.

2.2 Software development

A major focus of this code base is the implementation of modern software development practices, such as unit testing and inline documentation. We want OpenOA to inspire confidence from the community of wind industry users. As a step in this direction, we have unit tests throughout the toolkits module[2] and integration tests for the methods library[3]. These practices ensure that the code not only produces reliable results but is also traceable and maintainable for the lifetime of the project. Furthermore, to promote maintainability and readability from a broad user community, the OpenOA code was developed to be consistent with Python PEP-8 standards[4].

Tests are implemented using the Python *unittest* library and can be executed using *pytest*. Toolkit functions each have at least one unit test, whereas analysis methods have a fixed integration test to ensure consistency of results after changes in code. Unit tests are typically based on random data but may be based on real wind power plant data (e.g., power curve model testing). Integration tests are typically based on real data and are designed to ensure consistency in some calculated metric (e.g., operational AEP). The unit test coverage for the Toolkits is currently over 90% and we intend to expand and improve this coverage in future releases. Testing and validation will become more critical should more users within NREL and the broader wind energy community begin contributing to the code base.

Tests are run automatically through a continuous integration framework each time a developer wants to merge their changes with the main code base. This practice can prevent bugs from being accidentally committed to the code base. For code changes through the github.com repository (see Section 2.3), we use TravisCI[5].

Finally, OpenOA includes full web-published documentation using the Sphinx documentation platform[6]. The documentation is generated from inline docstrings, as well as reStructured Text (RST) files located in the *sphinx* folder of the repository. The documentation includes descriptions of every function and module in OpenOA and provides example analyses to help users get started with the software.

[2]Unit testing involves the regular testing of individual functions, or 'units', of software to ensure the software is performing as designed.
[3]Integration testing occurs after unit testing, and involves the testing of the combination of software modules as a group.
[4]https://www.python.org/dev/peps/pep-0008/
[5]https://travis-ci.org/
[6]http://www.sphinx-doc.org/en/master/

Wind Energ. Sci. Discuss., https://doi.org/10.5194/wes-2019-12
Manuscript under review for journal Wind Energ. Sci.
Discussion started: 2 April 2019

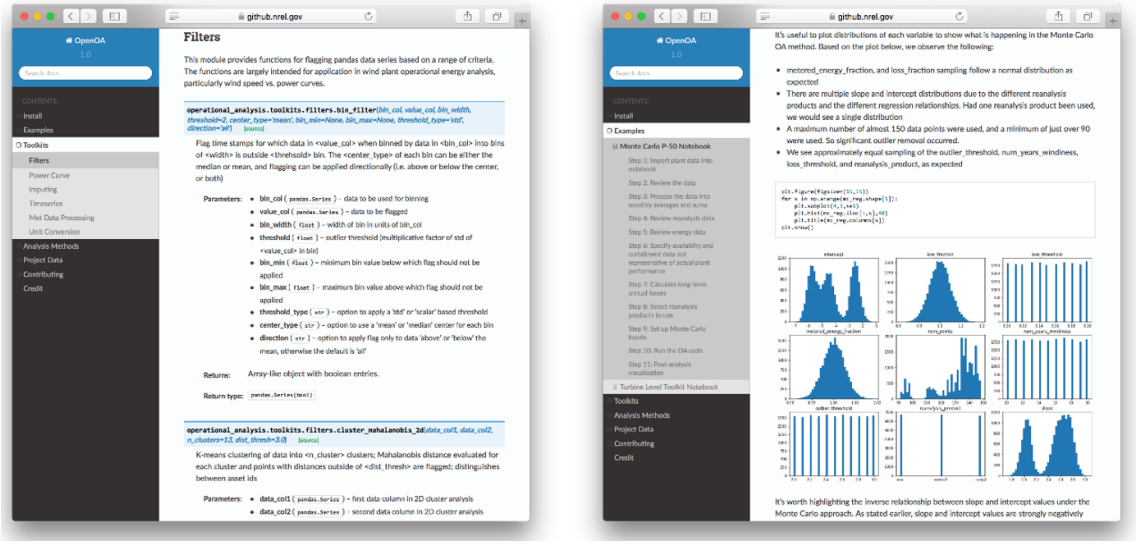

Figure 2. Screenshot of the documentation for OpenOA. Left: Inline documentation for the filters toolkit. Right: AEP calculation example.

2.3 OpenOA Software Repository

The OpenOA repository was developed using the git source code management platform and a public release is available on github.com, an open platform for collaborative development on software projects Laboratory (2018). The software development team at NREL followed standard git practices, with a production (master) release of the software supplemented by active development in a separate branch.

The public OpenOA repository currently contains a branch for the version 1.0 release. To submit code contributions, members of the public can fork the OpenOA repository, create feature branches in their own fork, and finally submit a pull request into the OpenOA repository through GitHub. In addition to pull requests, the core OpenOA developers will monitor the issue tracker associated with the OpenOA GitHub repository. Feedback, bug reports, and feature suggestions can be made through this issue tracker.

3 Description of version 1.0 functionality

This section provides an overview of the version 1.0 capabilities and applications of the Types, Toolkits, and Methods modules described in the previous section. More detailed documentation of these modules can be found on the GitHub repository Laboratory (2018).

Wind Energ. Sci. Discuss., https://doi.org/10.5194/wes-2019-12
Manuscript under review for journal Wind Energ. Sci.
Discussion started: 2 April 2019

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

3.1 Types module functionality

There are four classes within the Types module. The core of this module, as described in Section 2.1, is the Plant Data class. In addition to housing all the wind plant data attributes, this class also includes useful functions for importing raw plant data into a Plant Data object, checks to ensure Plant Data conform to expected schema (e.g., column naming conventions), and loading/saving Plant Data to file using flexible file formats. We believe this functionality will become invaluable to our users, and represents our first step toward an industrywide data exchange format.

The remaining classes are used largely in support of the Plant Data class. The Timeseries Table class provides a data structure in which the underlying data frame back end can be Pandas or Spark. This flexibility allows for OpenOA to handle both smaller data sets (i.e., Pandas) or very large data sets requiring distributed and parallel processing (i.e., Spark). The Reanalysis Data module allows storage of multiple reanalysis data products as Timeseries Tables within the same class. The Asset Data module contains a GeoPandas data frame that contains information about the turbines and met towers at the wind plant (e.g. location, rated turbine capacity).

3.2 Toolkits module functionality

There are currently seven different OpenOA toolkits, which are listed in Table 1 along with a general description of their functions. Toolkit modules range from those used for general data processing (flagging, imputation, unit conversion, and time series modules) and those specifically intended for wind plant data processing (meteorological data processing, power curve fitting, and plotting).

An example of toolkit use is shown in Figure 3. Here, several power curve models are fit to filtered wind speed and power data for a specific turbine. As shown in the figure, data from the Plant Data SCADA attribute and several toolkit modules are used to perform the flagging and removal of outlier data, the fitting of the power curve, and the plotting of results. The steps of this particular example are provided in detail as a Jupyter notebook[7] on the GitHub repository.

The value of toolkit modules lies in their generality. Each function was written to operate on array-like objects, such as Pandas Series, Data Frames, and NumPy Arrays. In this way, the toolkit modules can be applied in a variety of situations that are both internal and external to the OpenOA code base.

3.3 Methods module functionality

As described in Section 2.1, the methods are high-level analyses that are generally implemented as a class and compute metrics of scientific interest while relying on the lower-level toolkit functions for their implementation. Version 1.0 of OpenOA includes one method: the calculation of wind plant AEP using operational data. This method was created to support the WP3 benchmarking initiative (described in Section 1.4).

[7]https://jupyter.org/

Wind Energ. Sci. Discuss., https://doi.org/10.5194/wes-2019-12
Manuscript under review for journal Wind Energ. Sci.
Discussion started: 2 April 2019

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

Module	Description
Filters	Functions for flagging data based on a range of criteria.
Imputing	Functions for filling in null data with interpolated (imputed) values.
Meteorological data	Functions for calculating common meteorological variables used in wind resource analysis.
Time series	Functions for common time series analysis, including missing time-stamp identification and gap filling.
Unit conversion	Functions for common unit conversions in wind energy (e.g., power to energy).
Power curve	Functions to fit data to a specified wind turbine power curve model (including parametric and nonparametric forms) and to then apply the power curve to wind speed data.
Plotting tools	Functions to produce common wind resource and energy-related visualizations (e.g., wind rose).

Table 1. Toolkit modules in Version 1.0 of OpenOA.

The AEP analysis is based on an industry-standard approach in which monthly gross energy for the wind plant (reported energy at the revenue meter corrected for availability and curtailment losses) is related to a monthly long-term wind resource through a linear regression relationship. Calculation of AEP involves several steps:

- Processing of the revenue meter energy, loss estimates, and long-term reanalysis wind resource data
- Review of different reanalysis data for suitability
- Linear regression outlier detection and removal
- Flagging and removal of high-energy loss months
- Application of regression relationship of energy and wind resource to the long-term resource to calculate long-term gross energy
- Estimation of long-term AEP from long-term gross energy and expected future losses

Wind Energ. Sci. Discuss., https://doi.org/10.5194/wes-2019-12
Manuscript under review for journal Wind Energ. Sci.
Discussion started: 2 April 2019

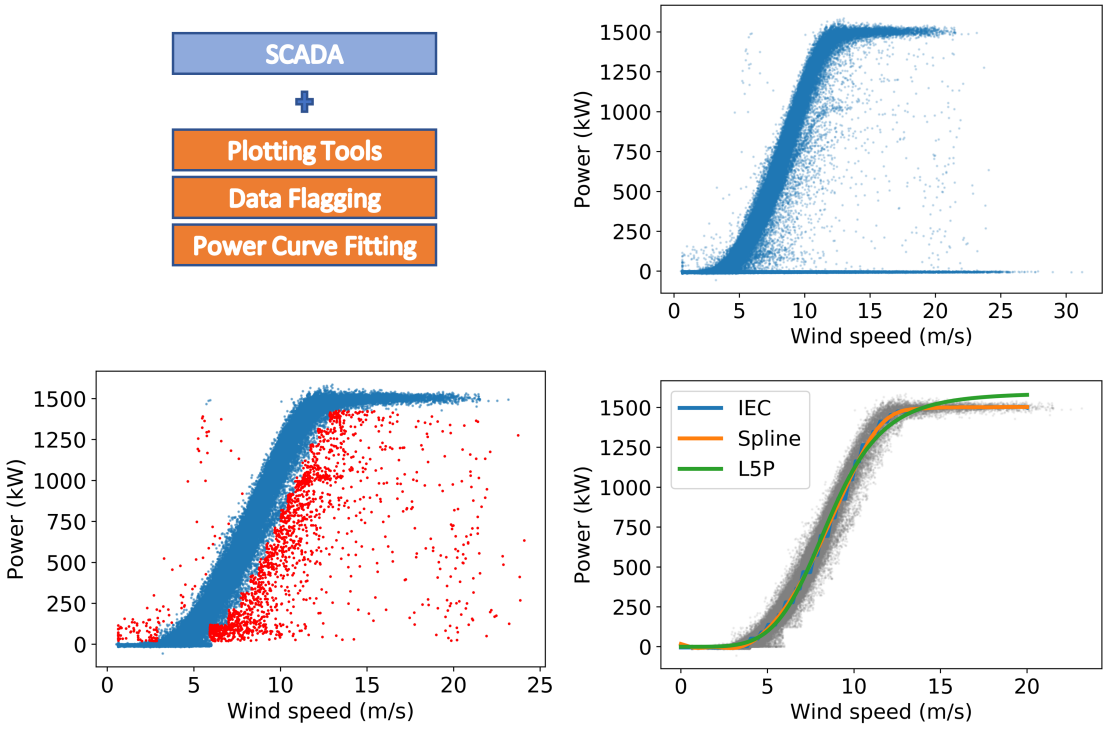

Figure 3. Using different OpenOA modules to calculate idealized power curves for a sample wind turbine. In this example, SCADA data is filtered and then fit using three different power curve models.

– Uncertainty quantification through a Monte Carlo approach in which inputs to and intermediate calculations within the process are sampled based on their assumed or calculated uncertainties.

An example usage of this method is shown in Figure 4. Here, revenue meter and reanalysis data attributes from Plant Data are used with several toolkit modules to calculate operational AEP for a wind plant. The details of this particular example are provided in a Jupyter notebook on the GitHub repository.

4 Toward version 2.0

Version 1.0 of OpenOA provides a foundational code base to support future scalability and a broad user and developer community moving forward. This foundation was established through the modular software architecture, extensive documentation, automated testing, and the use of modern software version control. However, the current functionality of OpenOA is somewhat

Wind Energ. Sci. Discuss., https://doi.org/10.5194/wes-2019-12
Manuscript under review for journal Wind Energ. Sci.
Discussion started: 2 April 2019

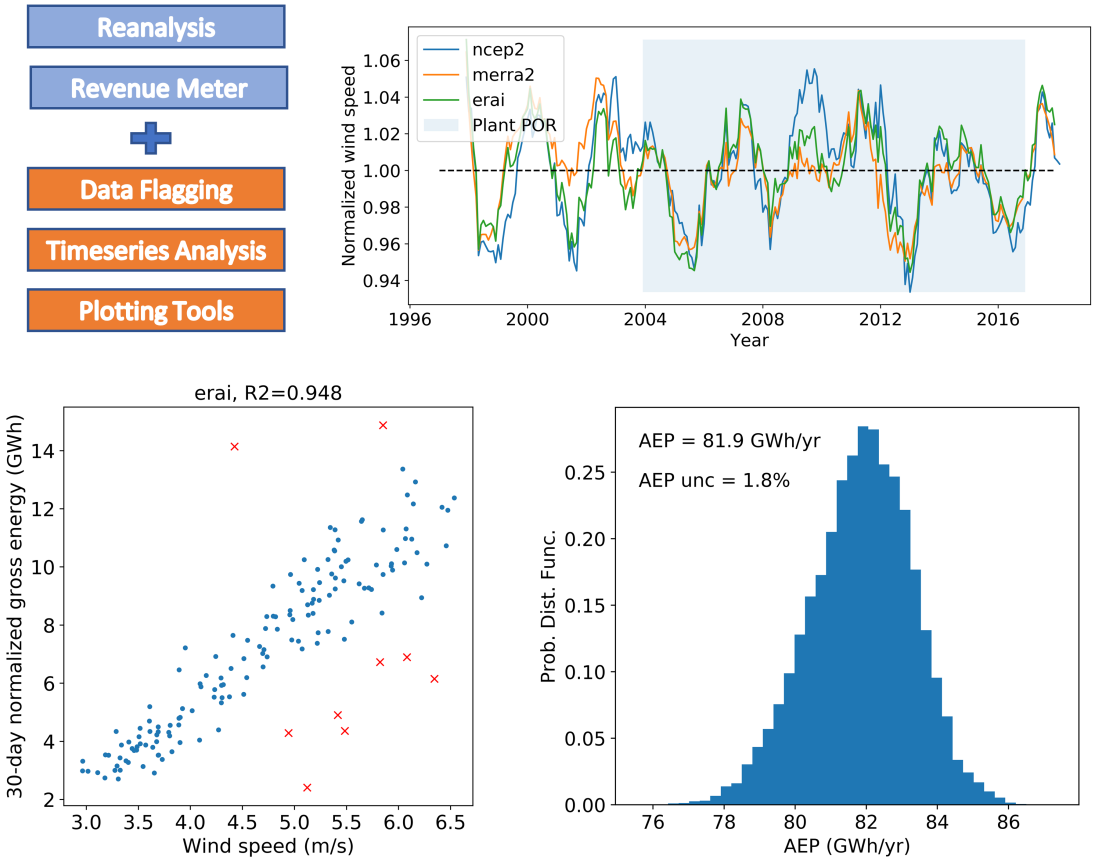

Figure 4. Using different OpenOA modules to calculate wind plant AEP using operational data. In this example, revenue meter and reanalysis data are processed using several toolkit modules.

narrow in scope, consisting mainly of lower-level processing capabilities (i.e., the toolkits) and the calculation of wind plant AEP (i.e., the only methods module), which were driven by the needs of the WP3 benchmarking project.

The road map toward version 2.0 focuses on three major goals: enhancement, expansion, and outreach. **Enhancement** will involve several adjustments to the code base that improve usability and scalability, but without the addition of new functionality or applications. **Expansion** will involve added functionality to the code base through new toolkits and methods designed for application to a wider range of wind energy problems. **Outreach** will involve engaging the wind energy community to become both users and developers of OpenOA moving forward. Specific development plans for each of these goals are described in detail in Table 2.

In terms of added functionality, NREL's focus toward version 2.0 will be the implementation of a gap analysis method for explaining the bias between an EYA-based and an OA-based AEP estimate. This framework would attribute the bias to sources

Wind Energ. Sci. Discuss., https://doi.org/10.5194/wes-2019-12
Manuscript under review for journal Wind Energ. Sci.
Discussion started: 2 April 2019

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

Goal	Description
Enhancement	Standardized data taxonomy conforming to IEC 61400-25
	Improved save/load functionality of Plant Data module
	Robust digital exchange format for Plant Data (e.g., Parquet files)
	ScikitLearn-style interface for AEP method
	Implementation of Spark option for SCADA processing
Expansion	Gap analysis workflow for comparing EYA- and OA-based AEP values
	Support for status code and event log processing
	Expanded long-term resource data sets (e.g., mesoscale models)
	Machine-learning module supporting multivariate, nonlinear analysis
Outreach	Host OpenOA tutorials via webinars
	Conduct one-on-one tutorials with interested users and developers
	Receive some external contribution to code base for version 2.0
	Gather stakeholder feedback on OpenOA road map

Table 2. Development and outreach roadmap for Version 2.0 of OpenOA

such as loss estimates (e.g., electrical, availability), wind resource, wind- and wake-flow modeling, and turbine performance (e.g., warranted vs. actual turbine power curve). Similar to the AEP method, development of this gap analysis framework will be done with strong stakeholder engagement and support.

We hope that by the time version 2.0 is released, multiple users external to NREL are using OpenOA and that one or more users have made some contribution to the code base (e.g., new toolkit function, new method). An ambitious outreach campaign will be conducted in 2019 in support of these goals.

5 Conclusion

Released publicly in September 2018, OpenOA provides an open-source framework for the operational analysis of wind power plant data. The intent of this effort is to begin fostering collaboration and methods sharing in a wind industry that has historically been very protective of methods and data. Much of an OA analysis could be standard and uncontroversial, and by creating a public repository for the collection and dissemination of OA methods and best practices, significant efficiency gains can be achieved. Furthermore, over time, we hope that OpenOA will become a reference implementation for OA methods from which a published standard (IEC or otherwise) may quickly follow. To our knowledge, this approach of first fostering a collaborative

repository of methods that are tested and used prior to developing a published standard is new to the wind industry, and may indeed prove more efficient than the multiyear approach of beginning immediately with a standard.

The development and release of OpenOA has strong wind industry backing. Current functionality in version 1.0 was strongly motivated by the Atmosphere to Electrons WP3 benchmarking study, and during the development process, most major stakeholders in the United States participated in the planning and review of the code base. The desire for an open-source solution to addressing wind energy problems is clear, and OpenOA provides the first framework for addressing that desire. It is our hope that in the near future, OpenOA evolves into the default code base used across the wind industry for performing operational analyses.

6 Acknowledgements

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.

Wind Energ. Sci. Discuss., https://doi.org/10.5194/wes-2019-12
Manuscript under review for journal Wind Energ. Sci.
Discussion started: 2 April 2019

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

References

Council, G. W. E.: Global Wind Report - Annual Market Update 2017, Tech. rep., Global Wind Energy Council, 2018.

Craig, A., Optis, M., Fields, M. J., and Moriarty, P.: Uncertainty quantification in the analyses of operational wind power plant performance, Journal of Physics: Conference Series, 1037, 052 021, http://stacks.iop.org/1742-6596/1037/i=5/a=052021, 2018.

IEC 61400-12-1:2017: Wind energy generation systems - Part 12-1: Power performance measurements of electricity producing wind turbines, Standard, International Electrotechnical Commission, 2017.

IEC 61400-15:draft: Assessment of site specific wind conditions for wind power stations, Standard, International Electrotechnical Commission, in draft.

IEC 61400-26-3:2016: Wind energy generation systems - Part 26-3: Availability for wind power stations, Standard, International Electrotechnical Commission, 2016.

Khatab, A. M.: Performance Analysis of Operating Wind Farms, Master's thesis, Uppsala University, Department of Earth Sciences, Campus Gotland, 2017.

Kusiak, A.: Renewables: Share data on wind energy, https://www.nature.com/news/renewables-share-data-on-wind-energy-1.19104, 2016.

Laboratory, N. R. E.: OpenOA, https://github.com/NREL/OpenOA, 2018.

Laboratory, N. R. E.: Wind plant performance prediction (WP3) benchmark report, Tech. rep., Department of Energy, in draft.

Lindvall, J., Hansson, J., Undheim, O., and Vindteknikk, J.: Post-construction production assessment of wind farms, Tech. Rep. 2016:297, Energyforsk, 2016.

Lunacek, M., Fields, M. J., Craig, A., Lee, J. C. Y., Meissner, J., Philips, C., Sheng, S., and King, R.: Understanding Biases in Pre-Construction Estimates, Journal of Physics: Conference Series, 1037, 062 009, http://stacks.iop.org/1742-6596/1037/i=6/a=062009, 2018.

Moseson, J.: Sharing real-time data will benefit all, https://www.windpowermonthly.com/article/1309619/sharing-real-time-data-will-benefit, 2014.

Wagner, R., Cañadillas, B., Clifton, A., Feeney, S., Nygaard, N., Poodt, M., Martin, C. S., Tüxen, E., and Wagenaar, J. W.: Rotor equivalent wind speed for power curve measurement – comparative exercise for IEA Wind Annex 32, Journal of Physics: Conference Series, 524, 012 108, http://stacks.iop.org/1742-6596/524/i=1/a=012108, 2014.