pvlib iotools—Open-source Python functions for seamless access to solar irradiance data

Jensen, Adam R.; Anderson, Kevin S.; Holmgren, William F.; Mikofski, Mark A.; Hansen, Clifford W.; Boeman, Leland J.; Loonen, Roel

Published in:
Solar Energy

Link to article, DOI:
10.1016/j.solener.2023.112092

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Jensen, A. R., Anderson, K. S., Holmgren, W. F., Mikofski, M. A., Hansen, C. W., Boeman, L. J., & Loonen, R. (2023). pvlib iotools—Open-source Python functions for seamless access to solar irradiance data. *Solar Energy*, *266*, Article 112092. https://doi.org/10.1016/j.solener.2023.112092

Solar Energy 266 (2023) 112092

Contents lists available at ScienceDirect

Solar Energy

journal homepage: www.elsevier.com/locate/solener

Data article

pvlib iotools—Open-source Python functions for seamless access to solar irradiance data

Adam R. Jensen [a,*], Kevin S. Anderson [b], William F. Holmgren [c], Mark A. Mikofski [c], Clifford W. Hansen [b], Leland J. Boeman [d], Roel Loonen [e]

[a] Department of Civil and Mechanical Engineering, Technical University of Denmark, Kgs. Lyngby, Denmark
[b] Sandia National Laboratories, Albuquerque, NM, USA
[c] DNV, Oakland, CA, USA
[d] Department of Hydrology and Atmospheric Sciences, University of Arizona, Tucson, AZ, USA
[e] Department of the Built Environment, Eindhoven University of Technology, Eindhoven, The Netherlands

ARTICLE INFO

Keywords:
Solar energy
Public data
Python
Data article
Free and open-source software (FOSS)

ABSTRACT

Access to accurate solar resource data is critical for numerous applications, including estimating the yield of solar energy systems, developing radiation models, and validating irradiance datasets. However, lack of standardization in data formats and access interfaces across providers constitutes a major barrier to entry for new users. pvlib python's iotools subpackage aims to solve this issue by providing standardized Python functions for reading local files and retrieving data from external providers. All functions follow a uniform pattern and return convenient data outputs, allowing users to seamlessly switch between data providers and explore alternative datasets. The pvlib package is community-developed on GitHub: https://github.com/pvlib/pvlib-python.

As of pvlib python version 0.9.5, the iotools subpackage supports 12 different datasets, including ground measurement, reanalysis, and satellite-derived irradiance data. The supported ground measurement networks include the Baseline Surface Radiation Network (BSRN), NREL MIDC, SRML, SOLRAD, SURFRAD, and the US Climate Reference Network (CRN). Additionally, satellite-derived and reanalysis irradiance data from the following sources are supported: PVGIS (SARAH & ERA5), NSRDB PSM3, and CAMS Radiation Service (including McClear clear-sky irradiance).

1. Introduction

The initial step in modeling solar energy systems typically involves obtaining weather data for the site of interest. In particular, it is important to obtain accurate and reliable information on the solar resource, as this is the most influential factor when estimating energy yield of solar energy systems. There are three primary sources for solar irradiance data: measurements from ground stations, reanalysis models, and satellite-derived datasets.

Within each category, there exist numerous data suppliers, encompassing both public and commercial datasets. To ease user access to the different data sources, efforts have been made to create standardized file formats. Most noticeable are the file formats for typical meteorological year (TMY) data, e.g., TMY2 [1], TMY3 [2], and EPW formats. In particular, the TMY3 format has become one of the most commonly used formats for TMY irradiance data, whereas the EPW file format has been adopted as the standard for building simulation tools. An attempt to develop a more flexible file format for solar irradiance and weather data was developed in the MESOR project [3]. However, the MESOR file format never gained widespread adoption. Overall, the aforementioned file formats have been plagued by several limitations (e.g., not permitting sub-hourly data or multiple measurements of the same quantity), which has regrettably led most data providers to develop their own distinct file structures and data retrieval methods.

Consequently, users are required to scrutinize each data provider's documentation and develop customized data solutions. This time-consuming process is in large part why identifying, accessing, cleaning, and exploring data can take up to 80% of the time in solar energy research projects [4]. An even more serious consequence is that this poses a major barrier to entry, discouraging users from exploring alternative datasets, often resulting in the use of sub-optimal data. In line with this, Gueymard [5] noted that "access to, and easy manipulation of large specialized atmospheric databases ... constitutes a serious limiting factor for most solar analysts".

For instance, the Baseline Surface Radiation Network (BSRN) serves as an illustrative example [6]. BSRN is recognized as the only global network dedicated to solar radiation measurements and is characterized by state-of-the-art infrastructure. However, obtaining data from

[*] Corresponding author.
E-mail address: arajen@dtu.dk (A.R. Jensen).

https://doi.org/10.1016/j.solener.2023.112092
Received 17 August 2023; Received in revised form 28 September 2023; Accepted 7 October 2023
Available online 3 November 2023

BSRN presents a significant challenge due to the use of a complicated, non-standard file format. This obstacle compels users to opt for lower-quality datasets or compromise spatial coverage, both of which are frequently encountered criticisms in solar energy research [7].

To enhance accessibility to solar resource data, a set of standardized functions has been developed in the Python programming language. The aim of this effort is to simplify the process and reduce the effort required for reading and retrieving solar irradiance data. The set of tools, called *iotools*, is part of the photovoltaic (PV) systems modeling package, pvlib python (henceforth referred to as pvlib) [8]. The name *iotools* derives from the term IO, which stands for Input/Output and refers to data transfer. This publication concerns the *iotools* subpackage contained in pvlib version 0.9.5 [9].

Specifically, the *iotools* subpackage is a collection of functions for reading local files and retrieving data from external providers. The functions have been developed such that they follow a similar and consistent pattern and return data in a uniform and structured manner. In this way, the time required for users to obtain and parse solar radiation data is reduced significantly. Going back to the example of the BSRN, data can be retrieved using the corresponding *iotools* function using only a single line of code, whereas the underlying code which parses the data spans 200 lines of code and 230 lines of documentation.

Related efforts to promote the accessibility of public solar datasets have been made in parallel with the development of pvlib *iotools*. Most noticeably, the *SolarData* R package was released in 2018 and featured functions for accessing five solar datasets [7]. The following year, an update to the *SolarData* package was released, adding functionality for accessing data from the BSRN [4]. Similarly, the *OpenSolar* R package [10] provides functions for accessing four different solar datasets, including data on solar irradiance, PV production, and electricity consumption. Another example is the Python package *IrradPy*, which provides functions for retrieving reanalysis data from MERRA-2 and clear-sky irradiance models [11]. Another initiative was recently launched by Blanc et al. [12], consisting of releasing quality-controlled solar irradiance ground measurements in the form of NetCFD files, following the GEO and FAIR principles. An overview of open-source tools related to PV modeling can be found at: https://openpvtools.readthedocs.io [13].

pvlib *iotools* offers access to 12 different solar irradiance datasets as of version 0.9.5. Additional benefits of the *iotools* subpackage include a proven long-term maintenance record (aiming for quarterly releases), adherence to documentation best practices, and standardized functions. In contrast, neither *OpenSolar* nor *IrradPy* has received significant updates since their initial release. Maintaining momentum in the long term is a major challenge faced by scientific software packages with only a few developers, where despite best intentions, competing priorities often cause maintenance and new development to drop off over time. Relative to the other mentioned packages, *iotools* association with the pvlib project means it benefits from the attention of the broader pvlib community, thereby bolstering long-term maintenance by decreasing the package's reliance on the continued availability of any single individual.

Notably, pvlib's large community of contributors (+100 contributors) also allows all contributions to be thoroughly peer-reviewed. In terms of usage, it can be mentioned the *iotools* documentation has seen 13.000 visits during the past year, accounting for 7% of visits to the pvlib documentation. The pvlib source code is available on GitHub: https://github.com/pvlib/pvlib-python. In addition to the pvlib-python package, the pvlib ecosystem also includes pvanalytics, a package for PV data quality assurance and feature recognition algorithms [14], and twoaxistracking, a package for simulating self-shading in arrays of two-axis solar trackers [15].

Additional to describing the *iotools* functionality and supported datasets, this article aims at providing information for users to make informed decisions on which datasets to use. The remaining part of this paper is structured as follows: the function pattern is described in Section 2, followed by a description of supported ground measurement datasets in Section 3 and modeled datasets in Section 4. Finally, a discussion of future work is described in Section 6.

2. Function pattern

The *iotools* functions adhere to a consistent, standardized pattern, aiming to make it effortless to read and retrieve weather data from different sources. The standardized pattern is unique to pvlib and encompasses several aspects, including:

- consistent naming and order of input parameters
- parsing date/time information and setting it as the index
- localizing time zones where applicable
- optional renaming of variables to a standard convention
- parsing metadata
- returning data structures in a uniform manner

This standardization ensures that there are minimal differences between the various functions, making it easy to switch between data providers or investigate alternative datasets.

As of pvlib 0.9.5 the *iotools* subpackage contains two categories of functions: `read` functions that read local data files (see Section 2.1) and `get` functions that retrieve external data (see Section 2.2). The function outputs are described in Section 2.3 followed by a description of the variable mapping in Section 2.4. Last, the documentation is discussed in Section 2.5 and inconsistencies in Section 2.6.

Listing 1: Illustration of the function patterns for the read and get-functions. The ellipsis represents possible additional parameters depending on the specific data provider.

```
# read function pattern
data, meta = read_x(filename, map_variables=True)

# get function pattern (measurement data)
data, meta = get_x(station, start, end,
                   username, password, ...
                   map_variables=True, url="https://...")

# get function pattern (modeled data)
data, meta = get_x(latitude, longitude, start, end,
                   username, password,  ...
                   map_variables=True, url="https://...")
```

2.1. Reading of local data files

Functions for reading local files are named `read_x`, where the *x* is replaced by the dataset name or abbreviation. For example, the function for reading data files from the BSRN is named `read_bsrn`. The main input to the read-functions is the `filename` parameter, which is a relative or absolute path to the local data file. An illustration of the `read` function pattern is shown in Listing 1.

The user interaction with the various file reading functions is almost identical for all file types. This is facilitated by the underlying code, which has been custom developed to parse the specific file type. Thus, the user avoids needing to understand the semantics of the various file formats, resulting in significant time savings.

2.2. Retrieval of external data

Retrieval of external data is achieved using the `get_x` functions. For data retrieval, there exist two function sub-categories, which depend on the type of the external dataset. The two dataset categories are (1) measurement data from ground stations and (2) modeled datasets covering a large geographical area (reanalysis or satellite-derived). The main difference in the function patterns is related to the location specifier (see Listing 1). For ground-measured data, only the station name or identifier is required to identify the location of interest. In contrast, for modeled irradiance dataset, the location is specified by the latitude and longitude as these datasets typically are gridded and have a large geographical coverage. The specification of latitude and longitude follows ISO 6709, i.e., latitude is defined as degrees north, and longitude is defined as degrees east. For both types of data retrieval functions, the desired time frame can be selected by specifying the `start` and `end` parameters.

Accessing certain external datasets may require credentials, e.g., a username and password or an API key. Information concerning how to obtain the necessary credentials is included within the function documentation as appropriate.

2.3. Outputs

Both the `read` and `get`-functions return two objects (see the left-hand side in Listing 1). The first object is the time series data, and the second object is the associated metadata.

The first element, `data`, contains the time series data in the form of a pandas DataFrame, which is a data structure containing tabular data similar to a spreadsheet. The `data` object contains several columns of data, typically global horizontal irradiance (GHI), air temperature, and other irradiance/weather-related parameters. Columns containing irradiance data are by default in the units of W/m^2. If the external service returns irradiation values (typically with units of Wh/m^2), these can be converted to irradiance values by specifying the `integrated` keyword (defaults to True). Furthermore, the timestamps in the data file are used to set the index of the DataFrame, making time series operations convenient. Whenever possible, the index is made time-zone aware (i.e., information on the timezone is assigned to each timestamp), which is required when calculating solar position. Users need to be aware that there is no standard practice as to whether timestamps refer to the start or end of the period (left or right bin labeling), which is an important consideration. Users are encouraged to consult the pvlib documentation and that of the data provider.

The second element, `meta`, is a dictionary containing the associated metadata. Metadata varies by data format and provider but commonly includes latitude/longitude, altitude above mean sea level, and time zone.

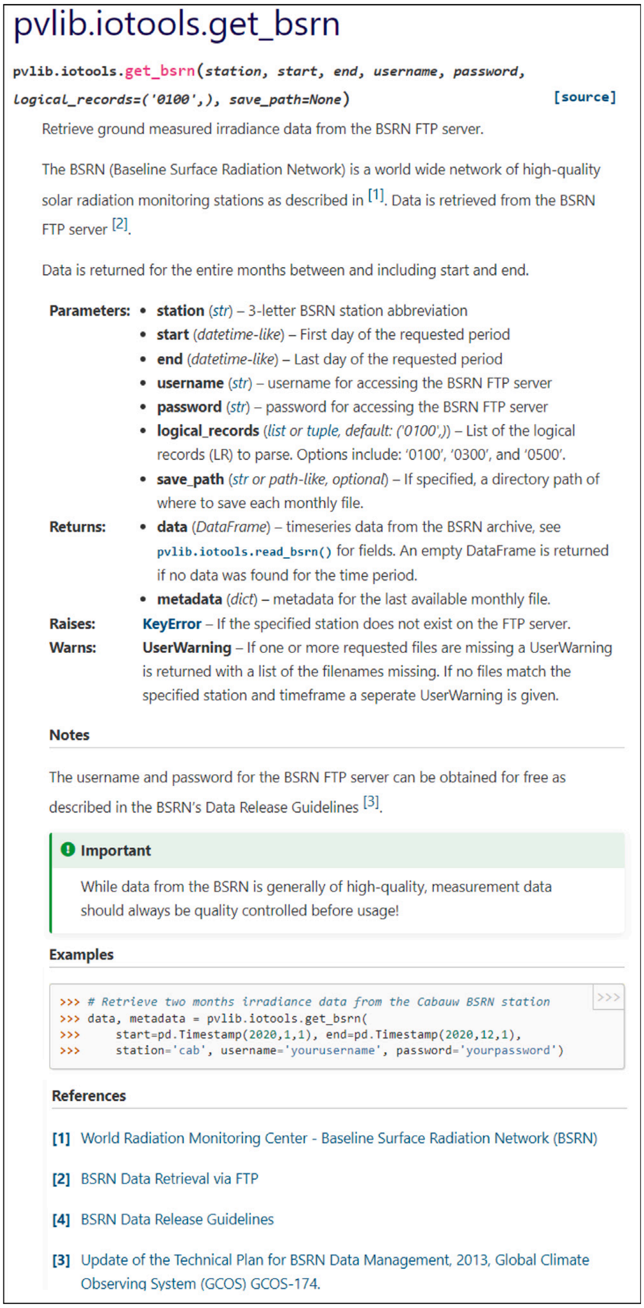
2.4. Variable mapping

Users have the possibility of renaming variables to pvlib's standardized naming convention by specifying the `map_variables` parameter. This is particularly convenient as data providers tend to use their own naming strategy, even for common parameters. For instance, global horizontal irradiance (GHI) may be named "Global Horiz" or "G(h)" for which the corresponding standard pvlib term is "ghi". Both the `read` and `get`-functions feature the `map_variables` parameter, which is set to true by default. Other notable standard variable names include "temp_air" for air temperature, "dni" for direct normal irradiance, and "dhi" for diffuse horizontal irradiance.

2.5. Documentation

All public functions in pvlib are thoroughly documented, which includes a description of the function and the input and output parameters. Additionally, the documentation also features links to references, typically a scientific publication or website providing additional information. Important supplementary information may also be specified in a separate notes section, such as the data frequency, geographical coverage, or procedures for obtaining credentials. The function documentation is contained within each function and adheres to the numpydoc style. The documentation is compiled using Sphinx by the "Read the Docs" platform and displayed at the pvlib website: https://pvlib-python.readthedocs.io.

An example of the documentation for the `get_bsrn` function is shown in Fig. 1. Note, the specifics of each function implementation and parameters are not presented here as this information is provided in the pvlib documentation.



pvlib.iotools.get_bsrn

pvlib.iotools.get_bsrn(*station, start, end, username, password, logical_records=('0100',), save_path=None*) [source]

Retrieve ground measured irradiance data from the BSRN FTP server.

The BSRN (Baseline Surface Radiation Network) is a world wide network of high-quality solar radiation monitoring stations as described in [1]. Data is retrieved from the BSRN FTP server [2].

Data is returned for the entire months between and including start and end.

Parameters:
- **station** (*str*) – 3-letter BSRN station abbreviation
- **start** (*datetime-like*) – First day of the requested period
- **end** (*datetime-like*) – Last day of the requested period
- **username** (*str*) – username for accessing the BSRN FTP server
- **password** (*str*) – password for accessing the BSRN FTP server
- **logical_records** (*list or tuple, default: ('0100',)*) – List of the logical records (LR) to parse. Options include: '0100', '0300', and '0500'.
- **save_path** (*str or path-like, optional*) – If specified, a directory path of where to save each monthly file.

Returns:
- **data** (*DataFrame*) – timeseries data from the BSRN archive, see `pvlib.iotools.read_bsrn()` for fields. An empty DataFrame is returned if no data was found for the time period.
- **metadata** (*dict*) – metadata for the last available monthly file.

Raises: **KeyError** – If the specified station does not exist on the FTP server.

Warns: **UserWarning** – If one or more requested files are missing a UserWarning is returned with a list of the filenames missing. If no files match the specified station and timeframe a seperate UserWarning is given.

Notes

The username and password for the BSRN FTP server can be obtained for free as described in the BSRN's Data Release Guidelines [3].

Important

While data from the BSRN is generally of high-quality, measurement data should always be quality controlled before usage!

Examples

```
>>> # Retrieve two months irradiance data from the Cabauw BSRN station
>>> data, metadata = pvlib.iotools.get_bsrn(
>>>     start=pd.Timestamp(2020,1,1), end=pd.Timestamp(2020,12,1),
>>>     station='cab', username='yourusername', password='yourpassword')
```

References

[1] World Radiation Monitoring Center - Baseline Surface Radiation Network (BSRN)

[2] BSRN Data Retrieval via FTP

[4] BSRN Data Release Guidelines

[3] Update of the Technical Plan for BSRN Data Management, 2013, Global Climate Observing System (GCOS) GCOS-174.

Fig. 1. Screenshot of the documentation of the `get_bsrn` function.

2.6. Inconsistencies

It should be noted that some functions deviate from the general pattern, as the pattern has evolved during the development of the package. The existing inconsistencies are mainly related to variable mapping and the returned data objects. For example, only a data object was returned for files that did not contain metadata, whereas newer functions also return an empty metadata object to follow the `(data, meta)` output pattern. Work is ongoing to update the existing functions that do not adhere to the standard function patterns.

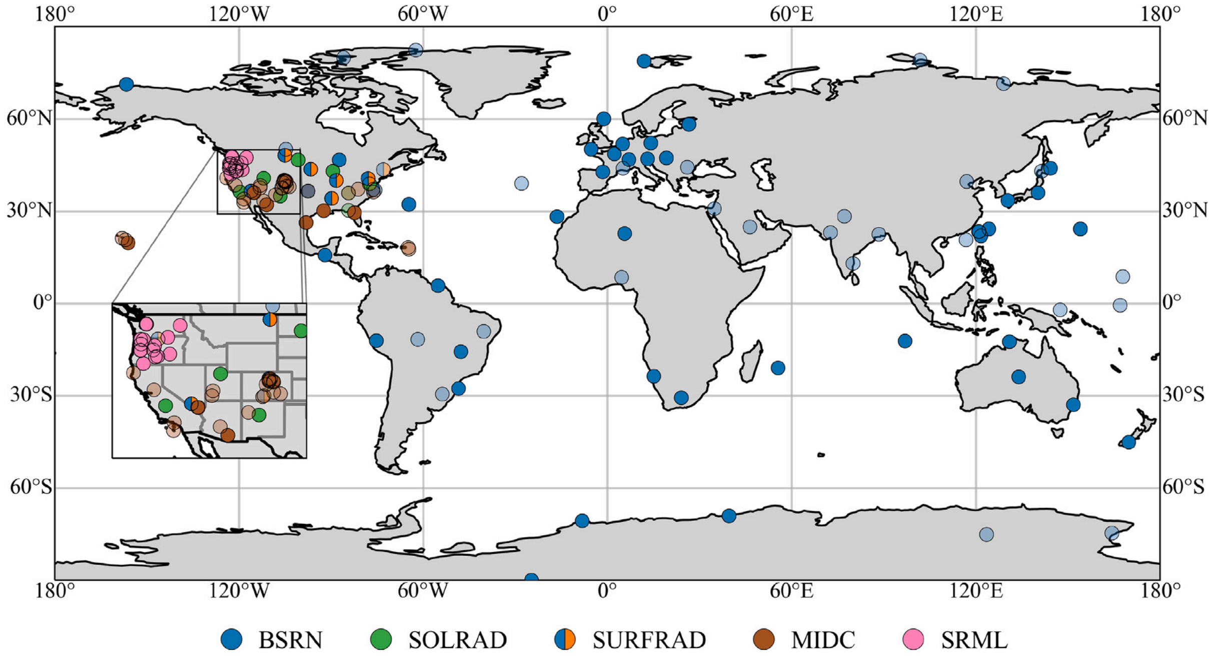


Fig. 2. Geographical distribution of ground stations. Each dot corresponds to a station, with the color corresponding to the station network (inactive stations as of July 2023 are shown as semi-transparent). For visualization purposes, CRN and inactive SRML stations are not shown. For an interactive map, see www.SolarStations.org.

3. Measurement irradiance data

The highest level of accuracy in assessing irradiance is achieved using well-maintained and quality-controlled ground-based instruments. Due to their high accuracy, ground-based measurements are critical for high-quality applications, e.g., benchmarking modeled irradiance datasets or developing empirical solar radiation models (e.g., [16,17]). Moreover, for large-scale solar energy projects, one-year ground measurement campaigns are recommended for 'tuning' longer time series of satellite-derived irradiance, a process known as site adaptation [18]. Another benefit of ground-measured irradiance is that measurements can be made at very high temporal resolution (sub-minute).

One drawback of ground measurements is the need for quality controlling the data and the existence of gaps in the data. Extensive quality control of measurement data is crucial as any long-term dataset inevitably contains some erroneous measurements. Some of the most common issues include instrument soiling (e.g., due to dust, pollen, or bird droppings), instrument misalignment, tracker malfunction, and humidity/frost. However, there is no universal method for quality-controlling irradiance data; the reader is referred to the recently developed comprehensive method by Forstinger et al. [16]. Due to the time-consuming maintenance requirements and high equipment costs, high-quality solar irradiance monitoring stations are scarce.

The following sections describe the ground station radiation networks for which the *iotools* supports data retrieval. A map of the supported ground stations is shown in Fig. 2 (with the exception of the CRN stations). For a global overview of solar radiation monitoring stations, the reader is referred to the online catalog www.SolarStations.org, which has been developed through the Assessing Solar initiative by members of the IEA PVPS Task 16 [19].

3.1. BSRN

The Baseline Surface Radiation Network (BSRN) is the only global network dedicated to monitoring solar radiation [6]. As of June 2023, the BSRN features 56 active stations, 1 candidate station, 13 inactive stations, and 6 stations pending status. The network provides all of its data freely, which can be obtained from an FTP server or www.pangaea.de. In order for a station to be included in the BSRN network, it must first demonstrate the ability to provide high-quality data for several years. Moreover, all BSRN stations are required to measure all three components of irradiance using thermopile radiometers and several additional meteorological parameters (including longwave irradiance), all at a 1-minute resolution.

One limitation of the BSRN lies in the utilization of an outdated file format known as the station-to-archive format. The file format splits each data entry across two lines, making the data difficult to read. However, this burden is removed by pvlib's `read_bsrn` and `get_bsrn` functions, which provide an easy way of reading and retrieving BSRN data. An example of data retrieval from the BSRN is provided in Listing 2. It should be noted that there is not a standardized time frame for which new data becomes available, thus, users are urged to check data availability at https://dataportals.pangaea.de/bsrn.

3.2. NREL MIDC

NREL's Measurement and Instrumentation Data Center (MIDC) provides irradiance and meteorological data from a network of stations across the United States. Unlike the BSRN, the MIDC does not have specific station requirements, e.g., not all stations have thermopile instruments. The time span and measurement interval also vary from station to station. As of July 2023, the MIDC network consists of 11 active and 23 inactive stations.

The most notable station in the MIDC network is the Baseline Measurement System (BMS) at the Solar Radiation Research Laboratory (SRRL) in Golden, Colorado. The BMS features the world's largest collection of radiometers in continuous operation [20]. This includes numerous pyranometers, pyrheliometers, pyrgeometers, spectroradiometers, and cloud cameras. Due to the large collection of continuously operated co-located instruments, the BMS data is ideal for comparing different types of instruments.

3.3. SRML

The Solar Radiation Monitoring Laboratory (SRML) is part of the University of Oregon and provides irradiance and meteorological data from a network of stations in the Northwestern United States. As of July 2023, there are 14 active stations and 24 inactive stations (see the inset map in Fig. 2). The SRML stations vary in quality and instrumentation, with some stations measuring all three irradiance components using high-quality thermopile instruments and others using rotating shadowband radiometers. All active stations record data at a 1-minute time resolution, although much of the historical data is only available at lower time resolutions. For a detailed description of the SRML network, see http://solardata.uoregon.edu and [21].

3.4. NOAA SOLRAD and SURFRAD

The National Oceanic and Atmospheric Administration (NOAA) operates two solar radiation monitoring networks in the United States. The first network is the SOLRAD network, which consists of nine stations and has operated since the mid-1990s [22]. The time resolution of the SOLRAD data has changed throughout the years, with measurements being made with a 1-minute frequency since 2015.

The second network is SURFRAD, which at its start in 1995 had four stations, which was expanded to six in 1998 and to seven in 2003 [23, 24]. The SURFRAD stations have been collecting measurements at a resolution of 1 min since 2009, while prior to that, they were logged at a resolution of 3 min.

Generally, SURFRAD stations are of higher quality and better maintained than SOLRAD stations. All SURFRAD stations are also part of the BSRN, although only the full number of parameters can be obtained from the SURFRAD FTP server. Both the SOLRAD and the SURFRAD networks measure all three irradiance components using thermopile radiometers, and the collected data is continuously quality controlled. The *iotools* subpackage currently only has a `read_surfrad` and `read_solrad` function and no dedicated `get`-functions. However, both `read`-functions support passing a URL as the `filename` parameter and thus are able to retrieve external data.

3.5. US climate reference network (CRN)

The US Climate Reference Network (CRN) was set up in 2004 with the aim of providing long-term measurements for climate research [25]. The network is managed by NOAA and consists of 114 sites in the contiguous United States, 21 stations in Alaska, and 2 stations in Hawaii. The primary variables measured at each site are air temperature, precipitation, surface temperature, and soil moisture and temperature. Solar irradiance is also measured but is considered an ancillary observation and is mainly used for quality-controlling air temperature measurements. The utilization of non-spectrally flat photodiode sensors and limited maintenance practices contribute to a considerably higher uncertainty level than stations specifically designed for solar radiation monitoring. As a result, users are strongly advised to exercise caution when employing irradiance measurements from the CRN network for solar energy applications. Data files can be read using the `read_crn` function, which also supports URLs for retrieval of external data.

3.6. Example use case

As previously mentioned, ground-measured irradiance data are necessary for evaluating the uncertainty of solar radiation models and datasets. A brief example is provided in Listing 2, demonstrating how to retrieve ground-measured irradiance data from the Cabauw BSRN station and compare measured and modeled direct normal irradiance (DNI). The modeled DNI is calculated using three different decomposition models and compared to measured DNI in Fig. 3.

Listing 2: Example of downloading of BSRN data and prediction of DNI using separation models.

```
import pvlib
import pandas as pd

# fetch measurements from the Cabauw BSRN station
data, meta = pvlib.iotools.get_bsrn(
    station=’CAB’,
    start=pd.Timestamp(2018, 7, 1),
    end=pd.Timestamp(2018, 7, 30),
    username=’username’, password=’password’)

# get solar position for each timestamp in "data"
solpos = pvlib.solarposition.get_solarposition(
    data.index , meta[’latitude’], meta[’longitude’])

# estimate DNI from GHI using the DISC model
data[’dni_disc’] = pvlib.irradiance.disc(
    data[’ghi’], solpos[’zenith’], data.index)[’dni’]
# estimate DNI from GHI using the Erbs model
data[’dni_erbs’] = pvlib.irradiance.erbs(
    data[’ghi’], solpos[’zenith’], data.index)[’dni’]
# estimate DNI from GHI using the DIRINT model
data[’dni_dirint’] = pvlib.irradiance.dirint(
    data[’ghi’], solpos[’zenith’], data.index)

# plot the measured and estimated DNI for three days
dnis = [’dni’, ’dni_disc’, ’dni_erbs’, ’dni_dirint’]
data[dnis].iloc[:4320].plot(ylabel=’DNI [W/m$^2$]’)
```

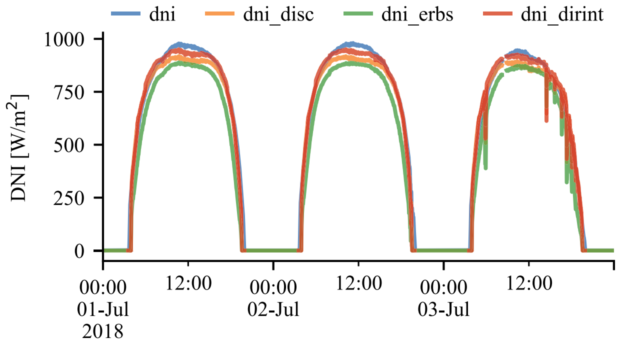


Fig. 3. Comparison of measured (blue line) and modeled DNI (output from Listing 2. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4. Modeled irradiance data

Most modeled irradiance datasets can be categorized as either satellite-derived or reanalysis. Modeled datasets are used at least in some stages of most solar energy projects due to several inherent advantages, despite having a higher uncertainty than what is achievable from ground-based measurements. The primary advantages of modeled irradiance datasets are large geographical coverage (continental scale), long time series (multiple decades), and low cost. Another advantage of modeled datasets is that they are typically serially complete, i.e., do not contain gaps.

Satellite-derived irradiance time series are based on cloud identification based on satellite images. In contrast, reanalysis datasets are derived from mathematical weather models combined with observational data. Generally, satellite-derived data tends to have a lower uncertainty and better spatial and temporal resolution than reanalysis datasets. However, reanalysis datasets are able to provide complete global coverage. In contrast, satellite-derived datasets are typically limited to ±60° latitude due to the view angle of geostationary satellites. It should be noted that satellite-derived irradiance models typically also rely on input data from reanalysis datasets, e.g., aerosols and precipitable water vapor.

As of pvlib 0.9.5, the *iotools* subpackage support six different modeled irradiance datasets from three data providers. A comparison of the temporal and spatial resolution and coverage of the different datasets

Table 1
Overview of the modeled irradiance datasets available through pvlib *iotools*. Data providers in parentheses denote the organization disseminating the data. Temporal resolution corresponds to the lowest available resolution (the number in parenthesis corresponds to the highest resolution of the atmospheric input data). The spatial resolution is approximate; see Fig. 4 for a detailed view of the spatial coverage.

Dataset	Provider	Type	Time frame	Temporal resolution	Geographical coverage	Nominal spatial resolution
CAMS All-sky Radiation	CAMS	Satellite	2004–	1-min (15-min)	Latitude: ±66° Longitude: −65° to 180°	Interpolated to point of interest
CAMS Clear-sky Radiation (McClear)	CAMS	Reanalysis	2004–	1-min (3-h)	Global	Interpolated to point of interest
SARAH-1	CM SAF (PVGIS)	Satellite	2005–2016	60-min	Latitude: −40° to 62° Longitude: −65° to 128°	5 km (0.05°)
SARAH-2	CM SAF (PVGIS)	Satellite	2005–2020	60-min	Latitude: ±65° Longitude: ±65°	5 km (0.05°)
ERA5	ECMWF (PVGIS)	Reanalysis	2005–2020	60-min	Latitude: −60° to 75°	25 km (0.25°)
NSRDB PSM3	NREL	Satellite	1998–2021	30-min 5-min from 2018	Latitude: −20° to 60° Longitude: −175° to 25°	4 km 2 km from 2018

is provided in Table 1. Additionally, the geographical coverage of the datasets is visualized in Fig. 4. All of the supported datasets are public, i.e., free to use. Each dataset is described in detail in the following sections, followed by a short example.

4.1. PVGIS

PVGIS is a web application allowing users to retrieve solar irradiance from various datasets and estimate PV system energy production [26]. Climatic variables, including temperature, wind speed/direction, and humidity, are also available and stem from the ERA5 and ERA5-Land reanalysis datasets. PVGIS is recognized as one of the most popular sources for irradiance data and PV production estimates and is an initiative by the European Commission's Joint Research Centre (JRC).

The most recent version is PVGIS 5.2, which was released in March 2022. As of July 2023, PVGIS supports four irradiance datasets, with the following three being described in the following sections: SARAH, SARAH-2, and ERA5. PVGIS also supports NSRDB PSM3, although accessing this dataset using the PSM3 API described in Section 4.2 is preferred (e.g., PSM3 data available through PVGIS is only available at 1-h resolution and for the period 2005 to 2015). Data from all datasets is also available for download as a TMY.

4.1.1. SARAH

The Surface Solar Radiation Data Set - Heliosat (SARAH) is a satellite-derived dataset of all-sky irradiance [27]. The SARAH data is derived from satellite observations from the geostationary Meteosat satellites and contain GHI, DNI, and effective cloud albedo. The dataset is developed by the EUMETSAT Climate Monitoring Satellite Application Facility (CM SAF).

PVGIS supports two versions of SARAH, SARAH-1 and SARAH-2. As seen in Fig. 4, the SARAH-2 dataset covers parts of South America, Europe, and Africa, whereas the older SARAH-1 dataset also covers Asia and parts of Australia. SARAH-2 is the successor of SARAH-1, and PVGIS urges users to use SARAH-2. A validation of SARAH-1 is presented in [28], and SARAH-2 is validated in [29].

4.1.2. ERA5

With the newest version of PVGIS, data is available for practically all inhabitable parts of the world due to the addition of ERA5 reanalysis data [30]. ERA5 is the flagship reanalysis dataset from the European Centre for Medium-Range Weather Forecasts (ECMWF) [31]. Due to the higher uncertainty compared to satellite-derived datasets, reanalysis data are primarily used in the polar regions.

4.2. NSRDB PSM3

The NSRDB (National Solar Radiation Database) is a collection of irradiance and meteorological datasets, the flagship of which is generated using the Physical Solar Model v3 (PSM3). The PSM3 estimates the three irradiance components using atmospheric data from various sources, including cloud properties from NOAA's GOES satellite imagery, aerosols from NASA's MERRA-2 and MODIS projects, and surface albedo from MODIS and the National Ice Center's IMS project. The NSRDB also provides other meteorological parameters, including ambient temperature, wind speed and direction, atmospheric pressure, and other values, based on data from MERRA-2. For a detailed description of the PSM3, see [32].

The NSRDB datasets are updated once annually, with data for the previous calendar year typically becoming available sometime in the second half of the year. The geographical coverage corresponds roughly to the contiguous US, Mexico, Central America, and northern South America. The 2-km and 5-min resolution is available from 2018 onwards. Historical data from 1998 onwards are available with a resolution of 4-km and 30-min. The NSRDB also supports downloading TMY data.

4.3. CAMS

The CAMS Radiation Service provides time series of global, diffuse, and direct irradiation at ground level, a product called CAMS All-Sky Radiation developed by the Copernicus Atmosphere Monitoring Service (CAMS). The geographical coverage is shown in Fig. 4.

The CAMS Radiation Service also provides clear-sky irradiation, denoted CAMS Clear Sky Radiation, derived using the McClear model. Both the all-sky and clear-sky irradiations are available with a time step of 1 min, 15 min, 60 min, daily and monthly, starting from 2004 to two days ago. The data is available in CSV and NetCDF format from the Copernicus Atmosphere Data Store (ADS) or SoDa. The pvlib function `get_cams` retrieves data using the SoDa web service [33]. The parameter `identifier` can be set to either `mcclear` or `cams_radiation` depending on which dataset is desired. Unlike most other datasets, data from the CAMS Radiation Service is not stored in a database but rather calculated on-the-fly; thus, only single-point requests are possible. It should be noted that the spatial resolution of CAMS only refers to the input data, which are then interpolated in time and space.

The McClear clear-sky model is a fully physical model based on the radiative transfer model libRadtran [34]. The model relies on inputs of aerosol properties, water vapor, and ozone from Copernicus. The ground-level irradiation is derived from the CAMS Clear Sky Radiation clear-sky irradiation time series coupled with satellite-based cloud information. The cloud information is extracted in a physical retrieval from images from the Meteosat Second Generation (MSG) satellite. As

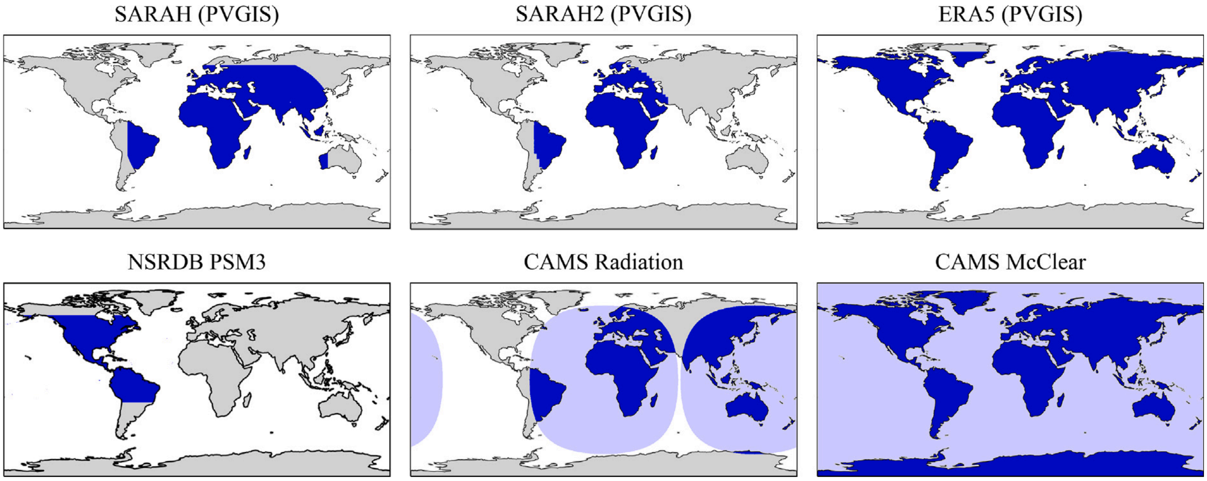


Fig. 4. Geographical coverage of the modeled irradiance datasets supported by pvlib *iotools*. Light blue indicates coverage over water. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of the second half of 2023, CAMS will also process images from the Himawari-8 satellite, thus adding coverage of Asia, Australia, and New Zealand as shown in Fig. 4. Irradiation is processed using the Heliosat −4 method [35,36].

The CAMS data is provided as irradiation in units of Wh/m^2; however, the pvlib `get_cams` function converts to irradiance (W/m^2) if the `integrated` keyword is set to true.

4.4. Example use case

Modeled irradiance datasets are often used when reliable ground station measurements are not available, for example, when estimating the annual output of a solar farm before the system is actually built. Listing 3 shows an example of retrieving a year's worth of weather data using the pvlib *iotools* subpackage, passing it through a basic PVWatts-style [37] PV system performance model built with other pvlib modules, and finally visualizing a subset of the simulated output power (Fig. 5).

Listing 3: Basic PV performance model using satellite-based irradiance

```
import pvlib

latitude = 35.0
longitude = -105.0

# fetch 5-minute data for all of 2019:
data, metadata = pvlib.iotools.get_psm3(
    latitude=latitude, longitude=longitude,
    api_key=’DEMO_KEY’, email=’user@example.com’,
    names=’2019’, interval=5)

# get solar position for each timestamp in "data"
solpos = pvlib.solarposition.get_solarposition(
    data.index, latitude, longitude)
# get tracker orientation for each timestamp
tracker_angles = pvlib.tracking.singleaxis(
    solpos[’zenith’], solpos[’azimuth’])
# transpose GHI, DHI, DNI into POA irradiance
components = pvlib.irradiance.get_total_irradiance(
    tracker_angles[’surface_tilt’],
    tracker_angles[’surface_azimuth’],
    solpos[’zenith’], solpos[’azimuth’],
    data[’DNI’], data[’GHI’], data[’DHI’])

# estimate PV cell temperature and power output
cell_temperature = pvlib.temperature.pvsyst_cell(
    components[’poa_global’], data[’Temperature’])
dc_power = pvlib.pvsystem.pvwatts_dc(
    components[’poa_global’], cell_temperature,
    pdc0=10000, gamma_pdc=-0.004)
# plot a few days as an example
dc_power.loc[’2019-12-15’: ’2019-12-17’].plot(
    ylabel=’DC Power [W]’)
```

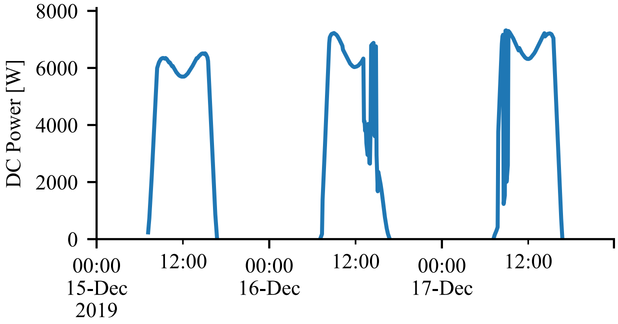


Fig. 5. A simple DC power prediction for a tracking system, using weather data retrieved from the NSRDB (output from Listing 3.

5. Licensing

The pvlib python code is open-source and released under a BSD 3-Clause license. It is up to the user to ensure that the usage of the data is used in accordance with the data provider's terms and conditions.

6. Outlook and future work

By providing users with a set of standardized and well-documented functions, pvlib *iotools* can significantly reduce the effort and time requirement of obtaining and working with solar irradiance data, in particular as all of the data providers discussed provide different access methods and file formats. Based on our experiences, an API type of data retrieval, as opposed to an FTP server for example, was found to be the fastest and easiest method for accessing data, and as a minimum, it is recommended that data is made available in CSV and JSON file formats. A REST API with keywords separated by ampersands can be recommended, which is a simple yet powerful and flexible solution. Moreover, several stakeholders are promoting the use of the NetCDF file format, which has several advantages but is not yet sufficiently widely adopted to be recommended as the only method of data access.

The pvlib community has expressed interest in functionality for accessing reanalysis data, which is particularly motivated by their global coverage and short time lag compared to public satellite-derived irradiance datasets (e.g., data from PVGIS and NSRDB is typically more than one year old). Specifically, work has been initiated to add functions for accessing the MERRA-2 and ERA5 reanalysis datasets. Additionally, there are ongoing efforts to add functionality for commercial providers of irradiance data (e.g., Solargis, SolarAnywhere, and Solcast). Furthermore, other public data sources which may be supported in the future include Atmospheric Radiation Measurement User Facility (ARM), Regional Test Centers (RTC), the SAURAN network, AERONET, and NASA POWER.

Furthermore, the *iotools* is not limited to irradiance data but should support the main data types used in PV performance modeling. With this in mind, it is planned to add functionality for retrieving precipitation data (for modeling soiling) and horizon profiles (for modeling far-shading). The *iotools* subpackage may in the future also contain functions for saving/writing data to specific file formats (e.g., export data in the SAM format). Additionally, efforts will be made to ensure that future functions adhere to the standard function patterns described in this paper. Lastly, it is important to mention that pvlib is an open-source library and is always welcome to new contributors.

CRediT authorship contribution statement

Adam R. Jensen: Writing – original draft (lead), Software. **Kevin S. Anderson:** Writing – original draft (supporting), Software. **William F. Holmgren:** Writing – review & editing, Software. **Mark A. Mikofski:** Writing – review & editing, Software. **Clifford W. Hansen:** Writing – review & editing, Software. **Leland J. Boeman:** Writing – review & editing, Software. **Roel Loonen:** Writing – review & editing, Software.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The author would like to acknowledge Afshin Andreas (NREL) for providing a list of MIDC stations, Josh Peterson (University of Oregon) for providing a list of SRML stations, Howard Diamond (CRN) for providing background information on the CRN network, and Amelie Driemel (BSRN data curator) for providing insight into the BSRN. Additionally, Marion Schroedter-Homscheidt (CAMS) was extremely helpful in providing additional information on the CAMS Radiation Service, and Nikos Alexandris (Joint Research Centre) provided data on the PVGIS geographical coverage.

Adam R. Jensen was supported by the Danish Energy Agency under grant numbers 64020-1082 and 134232-510237. Part of the work was also carried out through participation in the 2021 Google Summer of Code program.

Kevin S. Anderson and Clifford W. Hansen were supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Solar Energy Technologies Office Award Number 38267. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & 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. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy under the Solar Energy Technologies Office (SETO) and DE-FOA-0001649, Award Number DE-EE0008214.

References

[1] W. Marion, K. Urban, User's Manual for TMY2s (Typical Meteorological Years) - Derived from the 1961–1990 National Solar Radiation Data Base, Technical Report, Office of Scientific and Technical Information (OSTI), National Renewable Energy Laboratory (NREL), 1995, http://dx.doi.org/10.2172/87130.

[2] S. Wilcox, W. Marion, Users Manual for TMY3 Data Sets (Revised), Technical Report NREL/TP-581-43156, Office of Scientific and Technical Information (OSTI), National Renewable Energy Laboratory, Golden, Colorado, 2008, http://dx.doi.org/10.2172/928611.

[3] C. Hoyer-Klick, H. Beyer, D. Dumortier, M. Schroedter-Homscheidt, L. Wald, M. Martinoli, C. Schillings, B. Gschwind, L. Menard, E. Gaboardi, L. Ramirez-Santigosa, J. Polo, T. Cebecauer, T. Huld, M. Suri, M. de Blas, E. Lorenz, C. Kurz, J. Remund, P. Ineichen, A. Tsvetkov, J. Hofierka, Management and exploitation of solar resource knowledge, in: Proceedings of the EuroSun 2010 Conference, International Solar Energy Society, 2010, pp. 1–7, http://dx.doi.org/10.18086/eurosun.2010.13.10.

[4] D. Yang, SolarData package update v1.1: R functions for easy access of baseline surface radiation network (BSRN), Sol. Energy 188 (2019) 970–975, http://dx.doi.org/10.1016/j.solener.2019.05.068.

[5] C.A. Gueymard, Clear-sky radiation models and aerosol effects, in: J. Polo, L. Martín-Pomares, A. Sanfilippo (Eds.), Solar Resources Mapping: Fundamentals and Applications, Springer International Publishing, Cham, 2019, pp. 137–182, http://dx.doi.org/10.1007/978-3-319-97484-2_5.

[6] A. Driemel, J. Augustine, K. Behrens, S. Colle, C. Cox, E. Cuevas-Agulló, F.M. Denn, T. Duprat, M. Fukuda, H. Grobe, M. Haeffelin, G. Hodges, N. Hyett, O. Ijima, A. Kallis, W. Knap, V. Kustov, C.N. Long, D. Longenecker, A. Lupi, M. Maturilli, M. Mimouni, L. Ntsangwane, H. Ogihara, X. Olano, M. Olefs, M. Omori, L. Passamani, E.B. Pereira, H. Schmithüsen, S. Schumacher, R. Sieger, J. Tamlyn, R. Vogt, L. Vuilleumier, X. Xia, A. Ohmura, G. König-Langlo, Baseline surface radiation network (BSRN): Structure and data description (1992–2017), Earth Syst. Sci. Data 10 (3) (2018) 1491–1501, http://dx.doi.org/10.5194/essd-10-1491-2018.

[7] D. Yang, SolarData: An R package for easy access of publicly available solar datasets, Sol. Energy 171 (2018) A3–A12, http://dx.doi.org/10.1016/j.solener.2018.06.107, URL https://github.com/dazhiyang/SolarData.

[8] W.F. Holmgren, C.W. Hansen, M.A. Mikofski, Pvlib Python: A Python package for modeling solar energy systems, J. Open Source Softw. 3 (29) (2018) 884, http://dx.doi.org/10.21105/joss.00884.

[9] W. Holmgren, K. Anderson, C. Hansen, Calama-Consulting, M. Mikofski, A. Lorenzo, U. Krien, bmu, A.R. Jensen, C. Stark, A. Driesse, DaCoEx, M.S. de León Peque, kt, N. Priyadarshi, mayudong, Heliolytics, E. Miller, M.A. Anoma, V. Guo, L. Boeman, J. Stein, W. Vining, jforbess, T. Lunel, A. Morgan, J. Ranalli, S. Aneja, Carlosbogo, C. Leroy, pvlib/pvlib-python: v0.9.5, Zenodo, 2023, http://dx.doi.org/10.5281/zenodo.7748922.

[10] C. Feng, D. Yang, B.-M. Hodge, J. Zhang, OpenSolar: Promoting the openness and accessibility of diverse public solar datasets, Sol. Energy 188 (2019) 1369–1379, http://dx.doi.org/10.1016/j.solener.2019.07.016, URL https://github.com/fengcong1992/OpenSolar.

[11] J.M. Bright, X. Bai, Y. Zhang, X. Sun, B. Acord, P. Wang, Irradpy: Python package for MERRA-2 download, extraction and usage for clear-sky irradiance modelling, Sol. Energy 199 (2020) 685–693, http://dx.doi.org/10.1016/j.solener.2020.02.061, URL https://github.com/BXYMartin/Python-IrradPy.

[12] P. Blanc, R. Jolivet, L. Ménard, Y.-M. Saint-Drenan, Data Sharing of In-Situ Measurements Following GEO and FAIR Principles in the Solar Energy Sector, Technical Report, MINES Paris, PSL Research University, Centre O.I.E. - Centre Observation, Impacts, Énergie - MINES Paris, 2022, http://dx.doi.org/10.23646/AC2M-8504.

[13] W.F. Holmgren, C.W. Hansen, J.S. Stein, M.A. Mikofski, Review of open source tools for PV modeling, in: 2018 IEEE 7th World Conference on Photovoltaic Energy Conversion, WCPEC a Joint Conference of 45th IEEE PVSC, 28th PVSEC & 34th EU PVSEC, 2018, pp. 2557–2560, http://dx.doi.org/10.1109/PVSC.2018.8548231.

[14] K. Perry, W. Vining, K. Anderson, M. Muller, C. Hansen, PVAnalytics: A Python package for automated processing of solar time series data, in: PV Performance Modeling and Monitoring Workshop, 2022, URL https://www.osti.gov/biblio/1887283.

[15] A.R. Jensen, I. Sifnaios, K. Anderson, Twoaxistracking – a Python package for simulating self-shading of two-axis tracking solar collectors, MethodsX 9 (2022) 101876, http://dx.doi.org/10.1016/j.mex.2022.101876.

[16] A. Forstinger, S. Wilbert, B. Kraas, C.F. Peruchena, C.A. Gueymard, E. Collino, J.A. Ruiz-Arias, J.P. Martinez, Y.-M. Saint-Drenan, D. Ronzio, N. Hanrieder, A.R. Jensen, D. Yang, Expert quality control of solar radiation ground data sets, in: Proceedings of the ISES Solar World Congress 2021, International Solar Energy Society, 2021, http://dx.doi.org/10.18086/swc.2021.38.02.

[17] D. Tschopp, A.R. Jensen, J. Dragsted, P. Ohnewein, S. Furbo, Measurement and modeling of diffuse irradiance masking on tilted planes for solar engineering applications, Sol. Energy 231 (2022) 365–378, http://dx.doi.org/10.1016/j.solener.2021.10.083.

[18] J. Polo, C. Fernández-Peruchena, V. Salamalikis, L. Mazorra-Aguiar, M. Turpin, L. Martín-Pomares, A. Kazantzidis, P. Blanc, J. Remund, Benchmarking on improvement and site-adaptation techniques for modeled solar radiation datasets, Sol. Energy 201 (2020) 469–479, http://dx.doi.org/10.1016/j.solener.2020.03.040.

[19] A.R. Jensen, J.L. Lorente, P. Blanc, Y.-M. Saint-Drenan, AssessingSolar: An interactive guide to solar resource assessment in Python, in: Proceedings of the ISES Solar World Congress 2021, International Solar Energy Society, 2021, http://dx.doi.org/10.18086/swc.2021.37.03.

[20] A. Andreas, T. Stoffel, NREL Solar Radiation Research Laboratory (SRRL): Baseline Measurement System (BMS), Technical Report DA-5500-56488, National Renewable Energy Laboratory, Golden, Colorado, 1981, http://dx.doi.org/10.5439/1052221.

[21] J. Peterson, F. Vignola, Structure of a comprehensive solar radiation dataset, Sol. Energy 211 (2020) 366–374, http://dx.doi.org/10.1016/j.solener.2020.08.092.

[22] B.B. Hicks, J.J. DeLuisi, D.R. Matt, The NOAA Integrated Surface Irradiance Study (ISIS)—a new surface radiation monitoring program, Bull. Am. Meteorol. Soc. 77 (12) (1996) 2857–2864, http://dx.doi.org/10.1175/1520-0477(1996)077<2857:tnisis>2.0.co;2.

[23] J.A. Augustine, J.J. DeLuisi, C.N. Long, SURFRAD – a national surface radiation budget network for atmospheric research, Bull. Am. Meteorol. Soc. 81 (10) (2000) 2341–2358, http://dx.doi.org/10.1175/1520-0477(2000)081<2341:SANSRB>2.3.CO;2.

[24] J.A. Augustine, G.B. Hodges, C.R. Cornwall, J.J. Michalsky, C.I. Medina, An update on SURFRAD—the GCOS surface radiation budget network for the continental United States, J. Atmos. Ocean. Technol. 22 (10) (2005) 1460–1472, http://dx.doi.org/10.1175/jtech1806.1.

[25] H.J. Diamond, et al., U.S. climate reference network after one decade of operations status and assessment, Bull. Am. Meteorol. Soc. 94 (4) (2013) 485–498, http://dx.doi.org/10.1175/BAMS-D-12-00170.1.

[26] T. Huld, R. Müller, A. Gambardella, A new solar radiation database for estimating PV performance in Europe and Africa, Sol. Energy 86 (6) (2012) 1803–1815, http://dx.doi.org/10.1016/j.solener.2012.03.006.

[27] U. Pfeifroth, S. Kothe, J. Trentmann, R. Hollmann, P. Fuchs, J. Kaiser, M. Werscheck, Surface Radiation Data Set - Heliosat (SARAH) - Edition 2.1, Satellite Application Facility on Climate Monitoring (CM SAF), 2019, http://dx.doi.org/10.5676/EUM_SAF_CM/SARAH/V002_01.

[28] R. Urraca, A.M. Gracia-Amillo, E. Koubli, T. Huld, J. Trentmann, A. Riihelä, A.V. Lindfors, D. Palmer, R. Gottschalg, F. Antonanzas-Torres, Extensive validation of CM SAF surface radiation products over Europe, Remote Sens. Environ. 199 (2017) 171–186, http://dx.doi.org/10.1016/j.rse.2017.07.013.

[29] A. Gracia Amillo, N. Taylor, A. Martinez Fernandez, E. Dunlop, P. Mavrogiorgios, F. Fahl, G. Arcaro, I. Pinedo, Adapting PVGIS to trends in climate, technology and user needs, in: 38th European Photovoltaic Solar Energy Conference and Exhibition; 907-911, WIP, 2021, http://dx.doi.org/10.4229/EUPVSEC20212021-5BO.6.1.

[30] R. Urraca, A.M. Gracia-Amillo, E. Koubli, T. Huld, J. Trentmann, A. Riihelä, A.V. Lindfors, D. Palmer, R. Gottschalg, F. Antonanzas-Torres, Evaluation of global horizontal irradiance estimates from ERA5 and COSMO-REA6 reanalyses using ground and satellite-based data, Sol. Energy 164 (2018) 339–354, http://dx.doi.org/10.1016/j.solener.2018.02.059.

[31] H. Hersbach, B. Bell, P. Berrisford, S. Hirahara, A. Horányi, J. Muñoz-Sabater, J. Nicolas, C. Peubey, R. Radu, D. Schepers, A. Simmons, C. Soci, S. Abdalla, X. Abellan, G. Balsamo, P. Bechtold, G. Biavati, J. Bidlot, M. Bonavita, G. Chiara, P. Dahlgren, D. Dee, M. Diamantakis, R. Dragani, J. Flemming, R. Forbes, M. Fuentes, A. Geer, L. Haimberger, S. Healy, R.J. Hogan, E. Hólm, M. Janisková, S. Keeley, P. Laloyaux, P. Lopez, C. Lupu, G. Radnoti, P. Rosnay, I. Rozum, F. Vamborg, S. Villaume, J.-N. Thépaut, The ERA5 global reanalysis, Q. J. R. Meteorol. Soc. 146 (730) (2020) 1999–2049, http://dx.doi.org/10.1002/qj.3803.

[32] M. Sengupta, Y. Xie, A. Lopez, A. Habte, G. Maclaurin, J. Shelby, The national solar radiation data base (NSRDB), Renew. Sustain. Energy Rev. 89 (2018) 51–60, http://dx.doi.org/10.1016/j.rser.2018.03.003.

[33] ECMWF, CAMS radiation service, 2021, URL http://www.soda-pro.com/web-services/radiation/cams-radiation-service/info.

[34] B. Gschwind, L. Wald, P. Blanc, M. Lefèvre, M. Schroedter-Homscheidt, A. Arola, Improving the McClear model estimating the downwelling solar radiation at ground level in cloud-free conditions - mcclear-v3, Meteorol. Z. 28 (2) (2019) 147–163, http://dx.doi.org/10.1127/metz/2019/0946.

[35] Z. Qu, A. Oumbe, P. Blanc, B. Espinar, G. Gesell, B. Gschwind, L. Klüser, M. Lefèvre, L. Saboret, M. Schroedter-Homscheidt, L. Wald, Fast radiative transfer parameterisation for assessing the surface solar irradiance: The heliosat-4 method, Meteorol. Z. 26 (1) (2017) 33–57, http://dx.doi.org/10.1127/metz/2016/0781.

[36] M. Schroedter-Homscheidt, F. Azam, J. Betcke, N. Hanrieder, M. Lefèvre, L. Saboret, Y. Saint-Drenan, Surface solar irradiation retrieval from MSG/SEVIRI based on APOLLO next generation and HELIOSAT?4 methods, Meteorol. Z. 31 (6) (2022) 455–476, http://dx.doi.org/10.1127/metz/2022/1132.

[37] A.P. Dobos, PVWatts Version 5 Manual, Technical Report NREL/TP-6A20-62641, National Renewable Energy Laboratory, Golden, CO, 2014, http://dx.doi.org/10.2172/1158421.