OpenSolar: Promoting the openness and accessibility of diverse public solar datasets

Cong Feng, Dazhi Yang, Bri-Mathias Hodge, Jie Zhang

Abstract

Observational solar data is the foundation of data-driven research in solar power grid integration and power system operations. Compared to other fields in data science, the openness and accessibility of solar data is lacking, which prevents solar data science from catching up with the emerging trends of data science (e.g., deep learning). In this paper, OpenSolar, a package with both R and Python versions, is developed to enhance the openness and accessibility of publicly available solar datasets. The OpenSolar package provides access to multiple types of solar data, primarily from four datasets: (1) the National Renewable Energy Laboratory (NREL) Solar Power Data for Integration Studies dataset, (2) the NREL Solar Radiation Research Laboratory dataset, (3) the Sheffield Solar-Microgen database, and (4) the Dataport database. Unlike other open solar datasets that only contain meteorological data, the four datasets in the OpenSolar package also contain behind-the-meter power data, sky images, and solar power data with satisfactory temporal and spatial resolution and coverage. The overview, quality-control methods, and potential usage of the datasets, in conjunction with sample code implementing the OpenSolar functions, are described in this paper. The package is expected to assist in bridging the gaps between the research fields of solar energy, power systems, and data science.

1. Introduction

With the rapid development of sensor, wireless transmission, and network communication technologies, large amounts of data have been accumulated in the power and energy domain, which has brought benefits of (i) enhancing system stability & reliability, (ii) increasing asset utilization & efficiency, and (iii) better communications between customers and utilities (Zhou et al., 2016; Tu et al., 2017). Solar energy data is a core element of informed solar energy decision-making, such as target setting & policy making (Cox et al., 2018), resource assessment (Kleissl, 2013), investment (Ondracek et al., 2015), forecasting (Kleissl, 2018), and solar power integration & grid management (Pohekar and Ramachandran, 2004). Nevertheless, solar data public availability for research purposes lags behind other fields (e.g., image processing) due to several impediments, such as business/security concerns and institutional/personal inertia (Pfenninger et al., 2017).

Great efforts have been made to increase the transparency and availability of solar energy data (Pfenninger et al., 2018), which has lead to over 40 datasets worldwide (Sengupta et al., 2017).

To further facilitate solar data research, promoting the availability and accessibility of solar data sets is highly encouraged by the solar energy community (Yang et al., 2018). For example, some well-known solar datasets, like the Surface Radiation Budget Network (SURFRAD) (Augustine et al., 2000) and National Solar Radiation Data Base (NSRDB) (Sengupta et al., 2018), were developed and released to the public. Solar forecasts have also become accessible from online platforms, such as the IBM Physical Analytics Integrated Data Repository and Services (PAIRS) (Lu et al., 2016). Despite the wealth of publicly available datasets, it is still challenging for users to easily access and pre-process these datasets. To this end, packages and libraries of R and Python functions, two of the most popular languages in data science and machine-learning, have been developed. For example, Lamigueiro...
et al. (2018) developed an R package, *meteoForecast*, to provide access to several Numerical Weather Prediction services. Yang (2018) developed an R package, *SolarData*, to get access to meteorological solar irradiance datasets. *PVLIB*, a Python package developed by Sandia National Laboratories, provides various solar data and simulation functions (Holmgren et al., 2015). However, most of the datasets accessed through current packages only include climatic and meteorological data, which does not always meet the needs of the solar research community due to the lack of (i) solar power data, (ii) behind-the-meter photovoltaic (PV) data, (iii) time-synchronous heterogeneous data that can be used for solar coordination research, such as data from electrical vehicles, appliances, water and gas, and (iv) data in other formats, such as sky images, which offer opportunities to take full advantage of deep learning techniques.

In this paper, a new package, *OpenSolar* (version 1.0), which has both R and Python versions, is developed to increase the openness and accessibility of diverse publicly available datasets. Ten functions are provided in the package to download, quality control (QC), read in, preprocess, and model with the data. In this paper, all the code examples used to demonstrate the functions are written in R, which can be found in the Github repository. The *OpenSolar* package should first be installed and loaded before calling its functions. The installation code and examples of function calls in the R package are shown as:

```r
library(devtools)
install_github("UTD-DOES/OpenSolar")
# Load the package and list the functions
library(OpenSolar)
ls.str("package:OpenSolar")
```

(1) The datasets accessed by *OpenSolar* contain diverse variables, including meteorological variables, simulated solar power, behind-the-meter PV measurements, behind-the-meter appliance measurements, sky images, etc.
(2) The datasets have satisfactory spatial or temporal coverage and resolution (e.g., Dataport contains 1 min data and SPDIS covers the entire US).
(3) The functions are well-packaged so that it is easy to access and preprocess the datasets.
(4) Example use cases are provided in the package for demonstration and benchmarking purposes.

The remainder of the paper is organized as follows. The four datasets and relevant R code are described sequentially in Sections 2–5. Section 6 introduces the modeling and visualization of the machine-learning based short-term solar forecasting benchmarks. Concluding remarks and ideas for future work are given in Section 7. The code for example usage and the Python version of the package are detailed in the Appendices.

2. NREL Solar Power Data for Integration Studies (SPDIS) dataset

2.1. Overview

The NREL SPDIS dataset was originally created for large-scale renewable energy integration studies, including the Western Wind and Solar Integration Study (GE Energy, 2010; Lew et al., 2013; Miller et al., 2014) and Eastern Renewable Generation Integration Study (Bloom et al., 2016). The SPDIS dataset consists of 1-year (2006) data of 5020 utility scale/distributed PV locations, covering 47 states (excluding Alaska, North Dakota, and Hawaii) of the United States. The distribution and capacities of PV locations are visualized in Fig. 1. It is found that Florida (593), California (405), and Georgia (332) are the three states with the most PV locations in this dataset. The number of distributed PV locations is larger than centralized PV plants, which is illustrated from the PV capacity distribution.

The SPDIS has 5-min simulated PV power data and hourly 1-day-ahead (1DA) and 4-h-ahead (4HA) PV power forecasts. The PV power is generated based on irradiance from the NSRDB Physical Solar Model (PSM) V2.0 version, which was developed based on a satellite cloud cover model by the State University of New York (GE Energy, 2010). The 1DA forecasts were generated by 3TIER based on numerical weather prediction (NWP) simulations (Brower et al., 2009), while 4HA forecasts are produced by using a persistence of cloudiness method (Lew et al., 2013).

2.2. Potential usage

Created for solar power integration studies, SPDIS has an extensive spatial coverage. Based on a meteorological dataset (i.e., NSRDB), SPDIS provides solar power information with spatial diversity, which is necessary for large-scale solar power integration research. For example, SPDIS has provided opportunities for system-level operation studies (GE Energy, 2010; Lew et al., 2013; Miller et al., 2014; Bloom et al., 2016), power market design (Tewari et al., 2011), energy storage research (Pandžić et al., 2015; Su and Gamal, 2013), among other applications. SPDIS can also be used to develop solar forecasting methods, which

---

are critical for power system operations. SPDIS has been used to investigate solar forecasting by considering spatial hierarchy (Yang et al., 2017), temporal hierarchy (Yang et al., 2017), and ensemble forecasting (Yang and Dong, 2018). The generality of the dataset helps explore statistical characteristics of solar forecasting, such as forecast error metric development (Zhang et al., 2015), solar forecast error analysis (Zhang et al., 2014; Zhang et al., 2013), and solar forecast benefit assessment (Martinez-Anido et al., 2016). To the best of our knowledge, SPDIS has the largest spatial extent among publicly available solar power datasets, which is beneficial to large-scale solar power integration and general forecasting pattern assessment (Feng et al., 2019).

2.3. Data pre-processing

SPDIS can be downloaded by state through HTTPS connections. The easiest way is by clicking the corresponding download links from the NREL SPDIS website. A more efficient approach is performing bulk download through the URL of the dataset. In the OpenSolar package, the SPDIS.download() function provides a neat front end to efficiently batch download SPDIS data by state. The following code downloads, unzips, and filters data of PV locations in Alabama and Arkansas:

```r
# specify your data directory
root_data <- 'your data directory'
list_data <- SPDIS.read(root_data, name_st = 'Texas', list_files = NA, readal 1l)
# extract time series and location information
data_ts <- list_data[[1]]
data_linfo <- list_data[[2]]
head(data_ts[,1:3], n = 3)
## LocalTime Location1_Power[MW] Location2_Power[MW]
## 1 01/01/06 00:00 0 0
## 2 01/01/06 00:05 0 0
## 3 01/01/06 00:10 0 0

head(data_linfo, n = 3)
## Local Time Latitude Longitude Capacity [MW]
## 1 1 32.05 -94.15 95
## 2 2 32.05 -94.35 27
## 3 3 32.35 -94.25 122
```

2.4. Quality control

The SPDIS dataset has already been pre-processed by a two-stage QC: the NSRDB QC and PV output QC. The NSRDB QC is performed by (i) restricting solar zenith angle to be less than 80°, (ii) setting negative irradiance to be zero, (iii) excluding missing values, and (iv) determining sky condition by cloud types (Habte et al., 2017; Sengupta et al., 2018). In terms of PV output QC, PV output is simulated using PVWatts (Dobos, 2014), which takes sub-hourly variability, derating factors, PV system installation parameters, site selection, temporal trends, coincident relationships with wind and load, and forecastability into consideration. More details of the PV output QC can be found in (GE Energy, 2010).

3. NREL Solar Radiation Research Laboratory (SRRL) dataset

3.1. Overview

The NREL SRRL has been collecting continuous solar
measurements at NREL’s South Table Mountain Campus (longitude: 105.18°W, latitude 39.74°N, elevation 1,828.2 m), Golden, Colorado since 1981. More than 80 instruments have been installed, including pyranometers, pyrheliometers, pyrgeometers, photometers, and spectroradiometers, which compose the Baseline Measurement System (BMS).\textsuperscript{6} The collected data have both high temporal-resolution (varying with measuring time and device) and diverse features. In addition to meteorological and climatic data, the SRRL dataset also contains two sets of total sky images, which are taken by a Yankee Total Sky Imager (TSI-800) and an EKO All Sky Imager (ASI-16). Both sets of TSIs have sky image snapshots and cloud-analyzed images with 10-min resolution. The TSI-800 has captured TSIs with a resolution of 288×352 pixels since 2004-07-14, while the ASI-16 started to record TSIs with a higher resolution of 1536×1536 pixels on 2017-09-26.

3.2. Potential usage

The NREL SRRL dataset is recognized for its high quality, which has been used as the benchmarking standard to validate the quality of other datasets with measured or simulated data, such as NREL’s Surface Radiation Budget Network dataset, NOAA’s Integrated Surface Irradiance Study dataset (Anderberg and Sengupta, 2014), and the NSRDB (Habte et al., 2017). SRRL has also been used to verify the effectiveness of irradiance models (Vick et al., 2012), spectral distribution derivation (Myers, 2012), and transposition models (Xie et al., 2018).

Due to its wide range of measurements, SRRL has been extensively used in solar energy research, such as solar forecasting (Reikard, 2009), and solar energy statistical analysis (Lave and Kleissl, 2010; Kang and Tam, 2013), PV panel design optimization (Lave and Kleissl, 2011), solar energy and electrical vehicle coordination (Saber and Venayagamoorthy, 2010; Saber and Venayagamoorthy, 2011). Unique among publicly available datasets, the sky images in SRRL dataset provide special input features for sky image processing based solar forecasting, including pixel red blue ratio statistics (Feng et al., 2018;}

\textsuperscript{6}https://midcdmz.nrel.gov/srrl_bms/.
Feng and Zhang, 2018; Feng et al., 2017b), cloud coverage (Saade et al., 2014), and cloud classes (Zhen et al., 2015).

3.3. Quality control

The SRRL QC that has already been performed on the dataset is twofold, including sky image QC and ground-based measurement QC. The sky image QC consists of selecting the high-resolution sky image set and pre-processing sky images. For the ground based measurements, there are over 190 measured variables collected from 80+ devices, which are maintained under NREL’s SERI-QC methodology (Stoffel and Andreas, 1981).

Considering the higher resolution and more advanced quality-control algorithms, the ASI-16 sky images are included in this package. Two sky images are taken by the ASI-16 every 10 min, a normal exposed original image (named with an ‘_11_NE’ extension) and an underexposed original image (named with a ‘_12_UE’ extension). The angle offset of the original images is rectified so that the calibrated images (with ‘_11’ and ‘_12’ extensions, respectively) are in the same direction as a map, i.e., North at the top. The pixels with zenith angles larger than 70° are discarded from the images to avoid hazy sky and obstacle presence (Luiz et al., 2018). In addition to the original and calibrated images, two sets of cloud-detected images are also provided based on the blue/red and blue/green ratio (BRBG) algorithm and the cloud detection and opacity classification (CDOC) algorithm (Ghonima et al., 2012) (named with ‘BRBG’ and ‘CDOC’ extensions, respectively). The BRBG algorithm classifies cloud pixels by an RGB threshold. The CDOC algorithm employs a clear sky library to improve the classification (both thin cloud pixels and thick cloud pixels) accuracy and removes the pixels affected by sun glaring (shown as the black triangle around the sun in Fig. 2(f)). A set of 6 example images of each variety are shown in Fig. 2.

SERI-QC performs post-measurement quality and uncertainty assessment to ensure that the data is within established boundaries and follows plausible patterns (Maxwell et al., 1993). The SERI-QC has been widely applied for solar irradiance QC (Zell et al., 2015; Diagne et al., 2014; Gueymard and Wilcox, 2011), and the quality-assured SRRL dataset is often used as ground-truth to calibrate other datasets (Anderberg and Sengupta, 2014; Lave and Kleissl, 2010).

3.4. Data pre-processing

With the inclusion of rich climatic and meteorological variables, SRRL can support a number of different types of solar research. Therefore, a function called `SRRL.download()` in the OpenSolar package was developed to access the wealth of variables, collected from over 80 devices. The variable and device information can be found on the SRRL BMS instrument description webpage. In addition to numerical measurements, the `SRRL.download()` function can also download ASI-16 sky images. Numerical time series and sky images can be downloaded together or separately based on different uses, controlled by the `skyimg` and `tmseries` parameters. The sky images are zipped by day, which can be unzipped by the function. Redundant images, except for raw images and the CDOC-processed images, are removed from the disk if the boolean parameter `ifunique` is TRUE. The following code provides an example of downloading both time series measurements and sky images within a time range:

```
# specify directory to save files
root_data <- getwd()
# list files in the directory
list.files(root_data)
## character(0)
# download sky images and irradiance time series between the date range
date_start <- '2017-12-30'
date_end <- '2018-01-02'
SRRL.download(root_data, date_start, date_end, skyimg=T, tmseries=T, ifunzip=T, ifunique=T)
# check downloaded files
list.files(root_data)
## [1] "20171230"
## [2] "20180101"
## [3] "20171231"
## [4] "20180102"
## [5] "SRRL_measurement_timeseries.csv"
```

A large collection of sky image features have been proven beneficial to very-short-term and short-term solar forecasting, including pixel RGB statistics (Yang et al., 2014), cloud motion vectors (Chow et al., 2011), and cloud coverage (Chu et al., 2013; Fu and Cheng, 2013). The OpenSolar package provides availability and flexibility of sky image pre-processing through the `SRRL.read()` function. Five critical features of raw sky images and CDOC-processed images are extracted, which are mean, standard deviation, the second-order entropy of raw sky image pixel normalized red-blue ratios, along with thin cloud coverage, and opaque cloud coverage from CDOC-processed images. We encourage processing images with more advanced techniques, such as convolutional neural networks, to extract other information. Therefore, the raw images are loaded and stored in pixel arrays with pixel RGB values, if `returnRGB` is TRUE. The following code shows an example of pre-processed images at one timestamp:

```
# specify directory to save files
root_data <- getwd()
# list files in the directory
list.files(root_data)
## character(0)
# download sky images and irradiance time series between the date range
date_start <- '2017-12-30'
date_end <- '2018-01-02'
SRRL.download(root_data, date_start, date_end, skyimg=T, tmseries=T, ifunzip=T, ifunique=T)
# check downloaded files
list.files(root_data)
## [1] "20171230"
## [2] "20180101"
## [3] "20171231"
## [4] "20180102"
## [5] "SRRL_measurement_timeseries.csv"
```

A large collection of sky image features have been proven beneficial to very-short-term and short-term solar forecasting, including pixel RGB statistics (Yang et al., 2014), cloud motion vectors (Chow et al., 2011), and cloud coverage (Chu et al., 2013; Fu and Cheng, 2013). The OpenSolar package provides availability and flexibility of sky image pre-processing through the `SRRL.read()` function. Five critical features of raw sky images and CDOC-processed images are extracted, which are mean, standard deviation, the second-order entropy of raw sky image pixel normalized red-blue ratios, along with thin cloud coverage, and opaque cloud coverage from CDOC-processed images. We encourage processing images with more advanced techniques, such as convolutional neural networks, to extract other information. Therefore, the raw images are loaded and stored in pixel arrays with pixel RGB values, if `returnRGB` is TRUE. The following code shows an example of pre-processed images at one timestamp:

```
# specify directory to save files
root_data <- getwd()
# list files in the directory
list.files(root_data)
## character(0)
# download sky images and irradiance time series between the date range
date_start <- '2017-12-30'
date_end <- '2018-01-02'
SRRL.download(root_data, date_start, date_end, skyimg=T, tmseries=T, ifunzip=T, ifunique=T)
# check downloaded files
list.files(root_data)
## [1] "20171230"
## [2] "20180101"
## [3] "20171231"
## [4] "20180102"
## [5] "SRRL_measurement_timeseries.csv"
```

7https://midcdmz.nrel.gov/srrl_bms/.
8The second-order entropy, also known as Shannon’s entropy, measures how much information contained in the image.
4. Sheffield Solar – Microgen database

4.1. Overview

The Microgen Database (Microgen)\(^9\) is a dataset managed by Sheffield Solar, University of Sheffield, which collects PV data from over 7000 locations across the United Kingdom (Sheffield Solar, 2016). The PV generation data is uploaded to the Microgen database by PV system owners, and thus has various lengths and temporal resolutions across the different locations. The dataset includes both residential and commercial PV installations between 0.7 kilowattpeak (kWp) and 69 kWp with different orientations and tilt angles (Taylor et al., 2015). Few PV power generation datasets with such a large spatial coverage are publicly available (especially residential PV), primarily because of privacy concerns. We believe that the Microgen dataset provides valuable opportunities for data-driven research in both the solar energy and power system domains. The Microgen database has been subjected to several QC methods, and erroneous data has been isolated and corrected by Sheffield Solar (Taylor et al., 2015).

4.2. Potential usage

The two advantages of Microgen are: (i) having a large spatial coverage and (ii) having behind-the-meter PV generation data, which have led to a wide range of publications on spatial solar energy analysis and distributed power system research. More specifically, the dataset’s spatial coverage is beneficial to large-scale PV monitoring and characterization (Taylor et al., 2015; Colantuono et al., 2014), solar power spatial aggregation modeling (Lingfors and Widén, 2016), and spatial-temporal solar forecasting (Silva and Brito, 2018). Behind-the-meter PV generation data has been used for distributed energy storage operation optimization (Hassan et al., 2017), distributed energy storage financial and environmental benefit assessment (Jones et al., 2017), PV investment analysis (Leicester et al., 2016), and PV self-consumption evaluation (Leicester et al., 2016).

4.3. Data pre-processing

Sheffield Solar requires submitting a data request form to gain access to Microgen. Two CSV files will be sent by email once the data request is approved, containing time series data and metadata of a collection of PV generation locations. A subset of Microgen was obtained by the authors and is used in this paper for demonstration purposes. The subset includes 50 distributed PV systems in the Southeast UK, which is geospatially diverse for solar energy research. The obtained meta information consists of geospatial coordinates, elevation, azimuth, capacity, panel size, etc., which are shown in Fig. 3. It is observed that all the PV systems are residential, since the maximum capacity is 3.45 kWp. The 50 PV generation time series range from 2014-01-01 to 2014-12-31 with a 30-min resolution.

The cumulative generation time series data is stored in one column of the CSV file, which is a typical database structure. The other three columns, i.e., id, date, and time, are used to differentiate PV location, date, and time information, respectively. The `Microgen.read()` function in the OpenSolar package calculates 30-min PV generation based on the cumulative generation reading and structurizes the one-column values in a standard multi-column time series data frame, the code for which is:

```r
# specify directory to save files
root_data <- getwd()
# which image to process
timestamp <- '2018-01-01 12:00'
# define parameters and process the image
root_data <- 'your data directory'
list_read <- SRAL.read(timestamp, root_data, returnRGB=T, processraw=T, processadv=T)
# check the results
img_raw <- list_read[[1]]
dim(img_raw)
## [1] 1501 1501 3
raw_feature <- list_read[[2]]
raw_feature
## [1] -0.3028366 0.1775277 -3.8358866
img_adv <- list_read[[3]]
dim(img_adv)
## [1] 1501 1501 3
cloud_coverage <- list_read[[4]]
cloud_coverage
## [1] 0.2172586 0.0524676
```

\(^9\)https://www.microgen-database.org.uk.
5. The Dataport database

5.1. Overview

The Dataport database is one of the largest datasets of disaggregated customer energy data, managed by Pecan Street Inc. Its data is collected from 1446 houses (1346 in Texas, 57 in Colorado, 58 in California, 4 in Oklahoma, and 1 in Illinois), of which 318 houses have PV installations (as of October 15th 2018). The dataset was originally created to understand customer behaviour regarding new devices, information, and price signals of the smart grid (Rhodes et al., 2014). The electricity consumption data of the entire home and various numbers of appliances is collected with less than or equal to 1-min resolution, along with natural gas and water consumption data. The Dataport dataset has a university schema, which offers all available data to researchers at US universities. The large number of monitored houses, high temporal resolution, and various behind-the-meter device (including behind-the-meter PVs) measurements make Dataport an ideal dataset for solar energy research.

5.2. Potential usage

Dataport has been extensively utilized in power system demand-side management (Bai et al., 2016), load forecasting (Tascikaraoglu and Sanandaji, 2016), electric vehicle integration (Shin and Baldick, 2017; Munshi and Mohamed, 2018), water/gas usage analysis and their nexus with electricity (Xue et al., 2017; Nagasawa et al., 2018; Vitter and Webber, 2018). It also contributes to solar energy research, such as distributed solar energy storage (Fares and Webber, 2017), solar-powered microgrid design (Halu et al., 2016; Lainfiesta et al., 2018), PV sizing (Kazhamiaka et al., 2018) and integration (Deetjen et al., 2017).

5.3. Data pre-processing

There are three ways to approach the Dataport database: (i) via Spotlight, an Interactive Database Access tool, (ii) via the PostgreSQL client, and (iii) via outside client connections. Spotlight is able to export data by specifying the date range, table name, measured variables, and house IDs. Data is also accessible by any PostgreSQL client (e.g., pgAdmin) through SQL commands. Both approaches need to export and store data from other integrated development environments, which takes independent steps and extra effort. Two functions, namely, Dataport.meta() and Dataport.get(), are provided in the OpenSolar package to extract, download, filter, clean, and format data for PV-related research.

To fully take advantage of Dataport, table names and a list of houses with PV installations can be extracted through the Dataport.meta() function. In order to do that, the connection between R and the Dataport database is first set up with a PostgreSQL Database Interface. A username and a password are required to connect to Dataport, details of access to which can be found on the Dataport Advanced Access webpage. The metadata can be downloaded if function parameter metadownload is TRUE and the saving directory is specified. An example of using the Dataport.meta() function is shown in the sample code below. There are a total of 72 tables in the Dataport database, which are useful for PV and PV-wind/load/electrical vehicle/energy storage/water/gas coordination research. The metadata includes a collection of information, such as state, city, PV installation capacity, and house size. It is important to note that the installation of a PV meter does not ensure the successful monitoring of PV generation (which will result in NA elements in metadata and NA PV columns in the house measurement time series table/data frame). This issue will be tackled by metadata QC, which will be introduced in Section 5.4.

The Dataport.get() function fetches behind-the-meter measurements of a house, once the house_id parameter is specified. The measurement time series has several time resolutions, i.e., 1 h, 15 min, and 1 min, which can be determined by the time_resolution parameter. The Dataport.get() function also has a QC option, which will be introduced in Section 5.4.

However, it is surprising that this dataset has not yet been used for solar forecasting. We believe that Dataport is able to stimulate research in PV generation forecasting, especially spatial-temporal PV forecasting, behind-the-meter PV forecasting, net load forecasting, and PV forecasting benefit assessment.

# specify data directory
root_data <- 'your data directory'
# read in Microgen data
data_microgen <- Microgen.read(root_data)
# check data dimension
dim(data_microgen)
## [1] 17520 52 # the first two columns are date and time

# specify the data directory
root_save <- getwd()
# specify function variables
usr <- 'your user name' # can be found on the Dataport Advanced Access webpage
psw <- 'your password' # once the registration request has been approved
metadownload <- T
QC <- F
# get the table names in Dataport database and PV meta data
list_result <- Dataport.meta(usr, psw, QC, metadownload, root_save)
table_list <- list_result[[1]]
data_meta <- list_result[[2]]
# check results
length(table_list)
## [1] 72
tnow(data_meta)
## [1] 318

10 https://dataport.cloud.
5.4. Quality control

QC of Dataport is performed by the OpenSolar package, which can be divided into metadata QC and time series QC. The metadata documents the PV installation information, which cannot guarantee the existence of measurements. To avoid empty PV columns returned by the Dataport.get() function, the right metadata should be properly quality-controlled. Since there is no additional information to identify the success of actual monitoring, PV value existence check is performed for all of the 318 PV installed houses by setting the QC parameter as TRUE. An example of quality controlled metadata extraction is shown in the following code segment. There are 190 out of 318 PV-installed houses having PV measurements. It is also important to note that checking through the electricity table in the database takes non-negligible computational time (around 116 min on a laptop with an Intel Core i7 2.6GHz processor and a 16.0 GB RAM).

```r
# specifies function variables
usr <- 'your user name' #can be found on the Dataport Advanced Access webpage
psw <- 'your password' # once the registration request has been approved
house_id <- 8872
time_resolution <- 'hours' # three options: 'minutes', '15min', 'hours'
QC <- F
data_house <- Dataport.get(usr, psw, house_id, time_resolution, QC)

# head(data_house)
## dataid localhour use air1 gen grid
## 1 8872 2012-05-02 01:00:00 0.4432833 0.1409667 -0.0119667 0.4328333
## 2 8872 2012-05-02 02:00:00 0.3588167 0.1258000 -0.0121667 0.3588167
## 3 8872 2012-05-02 03:00:00 0.3662167 0.1259833 -0.0120833 0.3662167
## 4 8872 2012-05-02 04:00:00 0.3735333 0.1250667 -0.0120333 0.3735333
## 5 8872 2012-05-02 05:00:00 0.2743833 0.0876833 -0.0120500 0.2743833
## 6 8872 2012-05-02 06:00:00 0.1884500 0.0000000 -0.0143667 0.1884500
```

Fig. 3. Distribution map of PV locations of a microgen subset.

```r
# setting up computational time calculation
begtime <- Sys.time()

# specify function variables
usr <- 'your user name' #can be found on the Dataport Advanced Access webpage
psw <- 'your password' # once the registration request has been approved
metadownload <- F
QC <- T
list_result <- Dataport.meta(usr, psw, QC, metadownload, root_save)

nrow(list_result[[2]])
## [1] 190

# print computational time
runtime <- difftime(Sys.time(),begtime,tz, units = c( "mins"))
cat("Overall Computational time (mins):" , runtime, "\n")
## Overall Computational time (mins): 115.7839
```
The measurements of Dataport are examined by several QC thresholds (the details are contained in the required non-disclosure agreement), during which spurious data are removed. In this case, four additional QC criteria are applied to (i) filter out discontinuous variables, which are defined as more than 10% values missing, (ii) restrict PV values between 0 and the rated capacity, (iii) linearly interpolate NA values, and (iv) identify deleted rows and perform linear interpolation. QC flags indicate good data by 0, missing data by 1, out-of-bound data by 2, and deleted data by 3. Extracting data with and without QC is shown as follows:

```r
# specify function variables
usr <- 'your user name' # can be found on the Dataport Advanced Access webpage
psw <- 'your password' # once the registration request has been approved
house_id <- '8872'
time_resolution <- '15min' # three options: 'minutes', '15min', 'hours'
QC <- T
data_unQC <- Dataport.get(usr, psw, house_id, time_resolution, QC)
nrow(data_unQC)
## [1] 38784
range(data_unQC$gen)
## [1] 0.008000 4.527467
QC <- F
data_QC <- Dataport.get(usr, psw, house_id, time_resolution, QC)
nrow(data_QC)
## [1] 38780
range(data_QC$gen)
## [1] -0.04346667 4.52746667
```

6. Benchmark modeling for short-term solar forecasting

Short-term solar forecasting plays an important role in power system operations. The OpenSolar package provides a function, MLForecast(), to mitigate the trivial benchmark modeling process in solar forecasting. A total of 10 powerful and widely-used machine-learning models are provided in the function, including 3 support vector regression (SVR) models, 3 artificial neural networks (ANNs), 3 gradient boosting machine (GBM) models, and a random forest (RF) model (Feng et al., 2017a). The training and forecasting process of these 10 models can be realized with just one line of code once the data is formatted and cleaned. In addition, forecasting time horizon and training/testing data partition ratio are required by the function. A matrix with both the forecast and actual time series, and a forecast evaluation metric table are generated. A visualization function, MLVisual(), is also provided to visualize the forecasting results. MLVisual() takes the forecast results generated by the MLForecast() function and outputs actual and forecast time series and barplots of three key forecast error metrics:

7. Conclusion and future work

In this paper, a package with both R and Python versions, OpenSolar, was developed to promote the openness and accessibility of solar data. Specifically, there are four publicly available datasets included in the package, namely, (i) NREL Solar Power Data for Integration Studies dataset, (ii) NREL Solar Radiation Research Laboratory dataset, (iii) Sheffield Solar-Microgen database, and (iv) Dataport database. The four datasets provide diverse types of solar C. Feng, et al.


1377
research data, such as: behind-the-meter measurements, PV power, sky images, and meteorological variables. This paper has described how these datasets, their quality control, potential uses, and sample code scripts from the OpenSolar package can be used to download and pre-process the data. The OpenSolar package is expected to bridge the gap among the solar data science, power system research, and machine/deep learning domains, making different solar data types more readily available to both solar energy researchers and non-domain experts.

In the future, we will extend this research in the following directions: (i) including more datasets in the package, (ii) providing deep-learning-based example usage of the datasets, and (iii) setting up standard datasets and benchmarking models for data-driven modeling in solar energy.

Appendix A. Example use case 1: temporal reconciliation with the Microgen dataset

Power systems require solar forecasts with various time resolutions. However, forecasting solar power independently with different time resolutions suffers from aggregate inconsistencies. Regression estimator-based reconciliation methods have been verified to preserve consistency in the solar forecasting temporal hierarchy and provide more accurate solar forecasts (Yang et al., 2017). In this appendix, we implement the temporal reconciliation methods proposed by Yang et al. (2017) and reproduce similar results as a Microgen dataset example use case. The case studies are included in the R script named MicrogenExample.R. Details of the temporal reconciliation methods can be found in Yang et al. (2017).

Appendix B. Example use case 2: geographical reconciliation with the SPDIS dataset

Solar forecasting plays crucial roles in different geographical-level power system operations. For example, customer-level forecasts can be used for demand response, while system-level forecasts can be used for unit commitment and economic dispatch. Being provided by different vendors, inconsistencies also exist among solar forecasts with different geographical resolutions. Yang et al. (2017) regulated the inconsistent spatial-hierarchical solar forecasts by geographical reconciliation techniques. The SPDIS dataset is used to reproduce the empirical portion of Yang et al. (2017), which aims to reveal the potential usage of the solar data with a large geographical coverage. Results of the case studies can be generated by executing the R script named SPDISExample.R. Details of the geographical reconciliation methods can be found in Yang et al. (2017).

Appendix C. OpenSolar Python library

The OpenSolar Python library (OpenSolar-Python) contains the same 5 functions with the identical input and output formats. The installation of the OpenSolar-Python can be conducted by pip, the Python package installer, from the Github.11,12 The installation code is shown as:

Appendix D. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.solener.2019.07.016.

References


Lamigueiro, Oscar Perpinan, Almeida, Marcelo Pinho, Lamigueiro, Maintainer Oscar

11https://github.com/UTD-DOES/OpenSolar_Python
12https://zenodo.org/badge/latestdoi/173795145