

A Comparison of PV Power Forecasts Using PVLlib-Python

William F. Holmgren*, Antonio T. Lorenzo*, Clifford Hansen†

*Department of Hydrology and Atmospheric Sciences, University of Arizona, Tucson, AZ, 85721, United States

†Sandia National Laboratories, Albuquerque, NM, 87185, United States

Abstract—We used the open-source PVLlib-Python library to create PV power forecasts for a fleet of utility scale power plants and assessed their accuracies. PVLlib-Python allows users to easily retrieve standardized weather forecast data relevant to PV power modeling from NOAA models including the GFS, NAM, RAP, HRRR, and the NDFD. A PV power forecast can then be obtained using the weather data as inputs to the comprehensive modeling capabilities of PVLlib-Python. We used these models to benchmark the performance of the University of Arizona’s configuration of the Weather Research and Forecasting model. Standardized, open source, reference implementations of forecast methods using publicly available data may help advance the state-of-the-art of solar power forecasting.

Index Terms—forecasting, performance modeling, PV modeling, software

I. INTRODUCTION

PVLlib-Python is an open source toolbox for PV modeling [1], [2]. Holmgren *et. al.* developed a forecasting module for PVLlib-Python to help the PV modeling community create benchmark solar power forecasts [3]. In this paper, we use the PVLlib-Python forecasting tool to create hourly average PV power forecasts for a fleet of utility scale power plants and we compare the forecasts to observed plant generation. We compare the forecasts derived from NOAA weather models, including the GFS, NAM, and RAP with forecasts derived from a high resolution mesoscale model run by the University of Arizona.

II. IRRADIANCE FORECAST DATA

The most critical component of a PV power forecast is the forecast of global horizontal irradiance (GHI). A GHI forecast can be obtained directly from a weather model forecast or it can be inferred from a model’s cloud cover forecast [4]. The suitability of each method depends on the parameterizations of the model, the data availability of the model, and the temporal resolution of the desired PV forecast. Parameterization issues include the accuracy of the solar position equation and the impact of aerosols in their radiative transfer algorithms [5]. Model data availability and temporal resolution depends on the data source. The PVLlib-Python forecast module described by Holmgren *et. al.* [3] accesses forecast data from the Unidata THREDDS server. However, the Unidata THREDDS server currently only hosts the most recent 2 weeks of forecast model output. To study a longer period of time, we wrote a Python script to download the relevant point forecast data from the NOAA NOMADS THREDDS data service [6]. Table I describes the model variable data availability and temporal

TABLE I
IRRADIANCE AND CLOUD COVER MODEL FIELD DATA AVAILABILITY. FOR NOAA MODELS, THIS DATA ONLY REFLECTS AVAILABILITY ON THE NOMADS THREDDS SERVER. OTHER NOAA DATA SOURCES MAY CONTAIN ADDITIONAL DATA.

Model	GHI	DNI	Cloud cover
GFS, 0.5 deg.	3/6 hr mixed interval average	None	3/6 hr mixed interval average
NAM, 12 km	1 hr for 36 hrs 3 hr for 84 hrs	None	1 hr for 36 hrs 3 hr for 84 hrs
RAP, 13 km	None	None	1 hr instant
UA-WRF, 1.8 km	3 min instant	3 min instant	None

resolutions for the NOAA GFS, NAM, and RAP models on the NOMADS THREDDS service.

For the NOAA models studied in this paper, we use a model proposed by Larson *et. al.* [4] to calculate GHI from cloud cover forecasts:

$$\text{GHI} = (\text{offset} + (1 - \text{offset})(1 - \text{cloud_cover}))\text{GHI}_{\text{clear}} \quad (1)$$

where $\text{offset} = 0.35$, cloud_cover is the total cloud cover, and $\text{GHI}_{\text{clear}}$ is determined by PVLlib’s `clearsky.ineichen` function and PVLlib’s climatological Linke turbidity table. The DISC model is then used to calculate DNI and DHI. Here, we use default values for all functions, however, forecasters may tune the parameterizations to minimize forecast errors.

TABLE II
COMBINATIONS OF WEATHER MODEL AND IRRADIANCE PROCESSING ALGORITHMS STUDIED.

Name	GHI	DNI
GFS-CC	GFS Cloud cover + Larson	GHI + DISC
UA-DISC	WRF	GHI + DISC
UA	WRF + Aeronet	WRF + Aeronet
NAM-CC	NAM Cloud cover + Larson	GHI + DISC
NAM-GHI	NAM GHI	GHI + DISC
RAP-CC	RAP Cloud cover + Larson	GHI + DISC

The University of Arizona Department of Hydrology and Atmospheric Sciences runs a convective-permitting, 1.8 km

resolution configuration of the Weather Research and Forecasting (WRF) model [7] for operational forecasts of weather, solar power, and wind power in Arizona and New Mexico [8]. As an example of the utility of PVLlib-Python for creating benchmark forecasts, we will compare PV forecasts derived from the UA-WRF configuration to the PVLlib-Python forecasts. WRF versions 3.7 and 3.8 were used for this study. The model parameterization was adjusted throughout the year, but includes SBU-YLIN and Morrison microphysics schemes, and ACM2 and BouLac planetary boundary layer schemes [9]. UA-WRF namelists are available at [8]. Forecasts from the 0Z, 6Z, and 12Z GFS and NAM runs provide the initial and lateral boundary conditions for a 5.4 km resolution outer domain, which in turn provides initial and boundary conditions for the 1.8 km resolution domain. For this study, we analyzed WRF models initialized with 6Z GFS data.

The UA-WRF model generates 3 minute resolution forecasts of GHI, DNI, 2 m temperature, and 10 m wind speeds, among other variables. This configuration of WRF does not account for the impact of aerosols on irradiance, so we post-processed the model’s irradiance forecasts using measurements of the previous day’s average aerosol optical depth obtained from the Aeronet site in Tucson, AZ [10]. We calculated daily average broadband AOD, τ_{bb} , from AOD measured at 380 nm and 500 nm [11], and then computed modified DNI and GHI as $DNI = DNI_{wrf} \exp(-\tau_{bb}/\cos\theta_z)$ and $GHI = GHI_{wrf} \exp(-0.01/\cos^{0.4}\theta_z)$, where θ_z is the solar zenith angle. We also studied UA-WRF derived PV forecasts in which DNI and DHI were inferred from the uncalibrated GHI using the DISC model.

For all models, we linearly interpolate the model forecast data from its native resolution, shown in Table I, to 5 minute resolution. For the GFS, NAM, and RAP models we then apply equation 1 and the DISC model, as discussed above, to determine a forecast GHI, DNI, and DHI. For the NAM model, we also create forecasts directly from its GHI forecasts. Table II summarizes the combinations of weather model data and processing algorithms studied in this paper. Figure 1(top) shows the result of the 3 hourly cloud cover to 5 minute irradiance conversion for the half-degree 2016-01-06-6Z GFS forecast. Figure 1(bottom) shows the UA-WRF GHI forecast and DISC-generated DNI and DHI forecasts. This UA-WRF model was initialized using the same 2016-01-06-6Z GFS forecast as shown in Figure 1(top).

III. FORECASTING PV POWER

We created forecasts for six PV systems in Arizona. The systems included three single axis trackers, totaling 63 MW AC, and three fixed tilt systems, totaling 14 MW AC. Five of the six systems are located near Tucson, Arizona, and the sixth system is located near Kingman, Arizona. Of the five systems in Tucson, three are in the same forecast model grid box. All of the studied PV systems are smaller than a forecast model grid box.

We chose to use a simple PV model (NREL’s PVWatts [12]) in which the only required parameters are DC name-

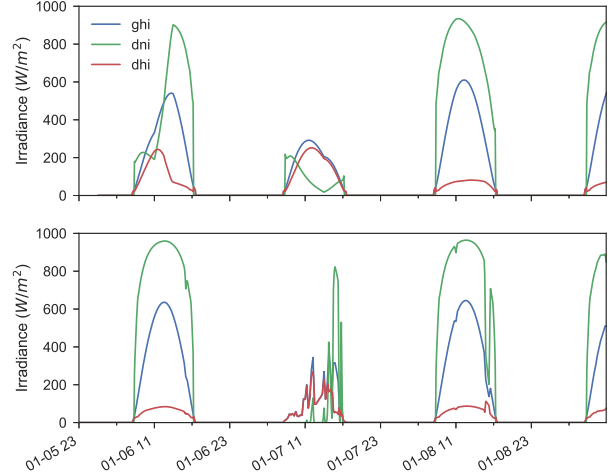


Fig. 1. 3.5 days of GHI (blue), DNI (green), and DHI (red) forecasts derived from a GFS forecast (top) and the UA-WRF initialized with the same GFS forecast (bottom) for Tucson, Arizona. GFS model irradiance is derived using equation 1 to determine GHI and the DISC model to determine its DNI and DHI. The UA-WRF model directly forecasts GHI and, for this figure, the DISC model was used to infer its DNI and DHI.

plate capacity and temperature coefficient. We imposed an additional maximum AC capacity parameter to account for inverter clipping. We determined system parameters by manually optimizing forecast model performance for clear days. This process was repeated for each forecast model to minimize the impact of forecast model temperature and wind speed biases. The optimum parameters did not vary by more than 10% from model to model. We resampled and processed the weather forecast data to 5 minutes, applied the PVLlib PV modeling tools, and compared 1 hour average forecasts and data. We used the Hay and Davies transposition model [13], PVLlib’s single axis tracker algorithm, a physical angle of incidence modifier model [14], and the Sandia Array Performance Model’s temperature model [15].

The code below demonstrates how the PV forecasts were made using the high-level, object-oriented PVLlib-Python API. Readers may consult the PVLlib-Python documentation [16] and source code [17] for more information on the function and method calls.

```
location = Location(
    latitude=32.2, longitude=-110.9, altitude=700)
system = SingleAxisTracker(
    module_parameters={
        'pdc0': 10, 'gamma_pdc': -0.0035})
system.peak_ac_power = 9
mc = ModelChain(
    system, location, orientation_strategy=None,
    dc_model='pvwatts', ac_model='pvwatts',
    aoi_model='physical', spectral_model='no_loss',
    temp_model='sapm', losses_model='no_loss')
fx_model = GFS() # or NAM or RAP
```

```

for fx_file in nomads_files:
    fx_data = pd.read_csv(fx_file)
    fx_data = fx_data.resample('5min').interpolate()
    fx_data = fx_model.process_data(fx_data)
    mc.run_model(fx_data.index, weather=fx_data)
    ac = mc.ac.clip_upper(system.peak_ac_power)
    ac = ac.resample('1h', label='right').mean()

```

The GFS-CC, NAM-CC, and RAP-CC classes' `fx_model` methods perform the following steps:

```

data['temp_air'] = self.kelvin_to_celsius(
    data['temp_air'])
data['wind_speed'] = self.uv_to_speed(data)
irrads = self.cloud_cover_to_irradiance(
    data[cloud_cover], **kwargs)

```

The cloud cover to irradiance method was described above. The NAM-GHI and UA-DISC processing algorithm is:

```

data['temp_air'] = self.kelvin_to_celsius(
    data['temp_air'])
data['wind_speed'] = self.uv_to_speed(data)
ghi = data['ghi']
solpos = self.location.get_solarposition(
    ghi.index)
dni = disc(
    ghi, solpos['zenith'], ghi.index)['dni']
dhi = ghi - dni * np.cos(
    np.radians(solpos['zenith']))

```

We used the `dask.distributed` framework to efficiently analyze the data in parallel. Additional processing code is available upon request.

IV. RESULTS

We studied forecast errors for all 2016 6Z GFS, 6Z NAM, and 9Z RAP models that were available on the online NO-MADS server. Half-degree GFS data was available for all of 2016, 12 km NAM data was available for August through December 2016, and 13 km RAP data was available for June through December 2016. We downloaded GFS data through 168 hours (out of a possible 384), NAM data through 72 hours (out of a possible 84), and RAP data through 18 hours (out of a possible 18 hours). These initialization times and time ranges ensured that an integer number of local sunrise through sunset periods was available for each model. The initialization times also ensured that data from these models would have been available by local sunrise of the first forecast day. Metered 1 minute resolution PV data was manually filtered for errors, and, where possible, scaled to correct for partial system outages. We focus our analysis on the accuracy of the forecast for all systems added together because this is often a more relevant number for a utility company that manages generation from a fleet of power plants. Hourly average forecasts derived from each day's weather models and the observed power are shown for four days in Figure 2.

We calculated absolute and normalized mean bias error, mean absolute error, and root mean squared error under many conditions, only some of which can be shown here. Additional information is available upon request. Errors were normalized

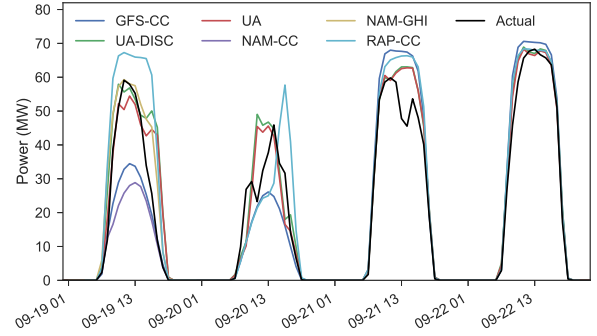


Fig. 2. Four days of PV observed generation (black) and forecasts (colors) derived from the GFS, NAM, RAP, and UA-WRF models using cloud cover (CC) or irradiance forecasts. Generation and forecasts are summed from 6 PV systems in Arizona.

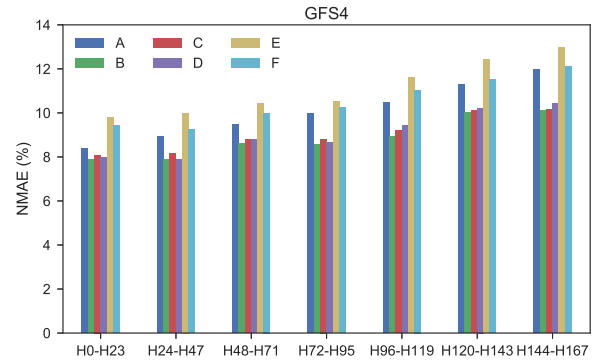


Fig. 3. NMAE forecast errors from the PVLlib-Python processed GFS forecasts for six PV systems (labeled A-F) as a function of forecast horizon. The time range for this analysis is Jan. 2016–Dec. 2016.

by the maximum AC generation observed for each plant during 2016.

First, we examined GFS forecast errors as a function of forecast day for each plant in the study period. Figure 3 shows that forecast errors grow as a function of forecast horizon for all systems. Other forecast models demonstrate similar trends for all systems and these trends serve as a sanity check of the algorithms. For different systems, NMAE for hours 0-23 ranges from 8%-10%, while NMAE for hours 144-167 ranges from 9%-12%. The remainder of this paper analyzes the aggregated generation and forecasts for all systems.

We compared the accuracy of all of the NOAA forecast models as a function of the forecast horizon, shown in Figure 4. Times at which any forecast was missing were removed from the comparative analyses. Therefore, the analysis of the NOAA models comprises most dates in August through December 2016. PV forecasts derived from NAM cloud cover and GHI have the lowest errors for hours 0-23, while forecasts derived from RAP cloud cover have the highest errors. The GFS and the two NAM forecasts have similar errors in

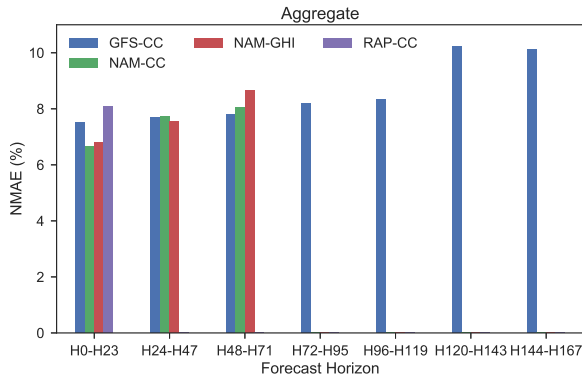


Fig. 4. NMAE forecast errors from the PVLlib-Python processed forecasts as a function of forecast horizon for the studied NOAA forecast models. The GFS-CC (blue), NAM-CC (green), and RAP-CC (purple) PV forecasts were derived from cloud cover forecasts, while the NAM-GHI (red) PV forecasts were derived from the NAM’s GHI forecast and the DISC model. The time range for this analysis is Aug. 2016–Dec. 2016. Times at which any forecast was missing were removed from the analysis.

hours 24-47, while the GFS slightly outperforms both NAM forecasts in hours 48-71. The cloud cover and GHI-based NAM forecasts perform similarly until forecast hours 48-71, at which the cloud cover based forecast is more accurate than the GHI based forecast. This is likely due to the fact that the NAM’s temporal resolution switches from hourly to 3-hourly at 36 hours, and the interpolation of the 3-hourly GHI data has significant errors.

Next, we use the GFS forecasts to benchmark the accuracy of the UA-WRF model initialized on the GFS data. The NMAE in Figure 5 shows that, for the systems studied here, the UA-WRF model’s dynamic downscaling of the GFS forecast yields a more accurate PV power forecast than the GFS under most forecast horizons and data subsets. The GFS forecast errors are similar for hours 0-23 and 24-47, and steadily increase beyond that. In contrast, the UA-WRF model’s hours 0-23 forecasts are more accurate than its hours 24-47 forecasts, but its hours 48-71 forecasts are no worse than its hours 24-47 forecasts. There is little difference in accuracy between the UA-WRF forecasts based on GHI and the DISC model, and UA-WRF forecasts based on Aeronet-corrected DNI and GHI.

Finally, we examined forecast accuracy as a function of month of year. Figure 6 shows the accuracy of each method for each month. The model errors exhibit similar trends, with some outliers. For most models, forecast accuracy is worse June through September, and best in May and November. These results led us to examine the relationship between forecast accuracy and clear sky condition. We downloaded irradiance observations from the NREL OASIS station located at the University of Arizona [18]. We used PVLlib-Python’s detect_clear function to determine if a minute is clear or not, summed the number of cloudy minutes in each month, and normalized this by the number of minutes with GHI greater than $1 W/m^2$. Figure 7 shows NMAE versus the percentage

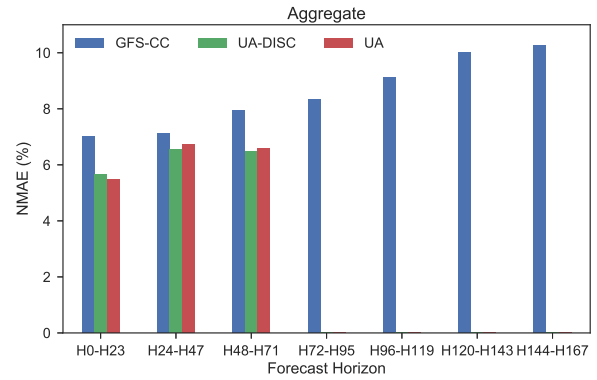


Fig. 5. NMAE forecast errors from the PVLlib-Python processed forecasts as a function of forecast horizon for the GFS with GHI derived from cloud cover (blue), UA-WRF with DNI derived from DISC (green), and UA-WRF with DNI post-processed with Aeronet data (red). The UA-WRF model was initialized with the GFS forecasts. The time range for this analysis is Jan. 2016–Dec. 2016. Times at which any forecast was missing were removed from the analysis.

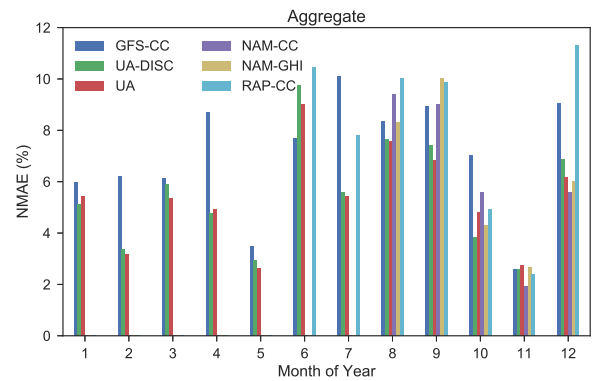


Fig. 6. NMAE intraday forecast errors from the PVLlib-Python processed intraday forecasts of each model over 2016.

of cloudy minutes per month. Relatively clear months, such as February, May, October, and November 2016 have lower errors, especially for the UA-WRF model.

V. FUTURE WORK

The emphasis of this work is to illustrate the use of PVLlib-Python to facilitate comparison and benchmarking of forecasts. The benchmarks suffer due to the restricted data availability of the NOMADS THREDDs service. NAM and RAP data were only available for half of 2016. The RAP model’s GHI variable was not available on the archive. Furthermore, the High Resolution Rapid Refresh model was not available on the NOMADS THREDDs service. A more comprehensive forecast archive would enable more accurate comparisons of forecast accuracy.

It is likely that the benchmark forecasts can be improved with modest effort. The GFS data used here is from the half-degree model. The newer quarter-degree GFS model may yield more accurate predictions, particularly at longer forecast

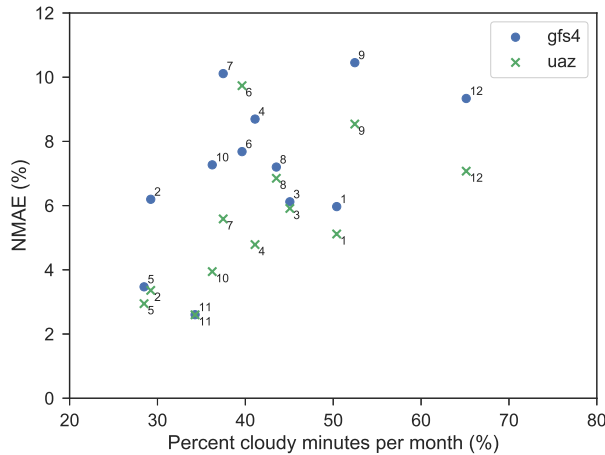


Fig. 7. Aggregate NMAE intraday forecast errors vs. percentage of cloudy minutes per month in Tucson, AZ. Points are labeled by month of year. Forecast errors are correlated with non-clear conditions.

horizons. A bias correction algorithm with a rolling training period could reduce seasonal trends in forecast skill. Scaling clear sky GHI by a different amount for low, mid, and high level cloud cover, rather than one scaling factor for total cloud cover, may yield significant improvements. Forecasts of aerosol optical depth and precipitable water could be used in the clear sky model for GHI.

VI. CONCLUSION

We used the PVLlib-Python forecasting module to compare the accuracy of solar power forecasts for a fleet of PV power plants. The PVLlib-Python library enables users to easily access weather forecasts and process them into PV power forecasts. The tool creates a standard set of data for PV modeling from the mixed data types provided from weather models. This work supports the standardization of PV power forecast methods, simplification of the use of weather forecast data for PV modeling, and fair and transparent PV power forecast model performance evaluation. As an example, we used PVLlib-Python forecasts to benchmark the accuracy of the UA-WRF model.

The PVLlib-Python documentation [16] provides examples for how to use the forecasting tool along with general PV system modeling. Readers are encouraged to participate in the PVLlib-Python community via its GitHub page [17] and the pvlib tag on stackoverflow.com.

ACKNOWLEDGMENTS

We gratefully acknowledge the Electric Power Research Institute, Tucson Electric Power, Southern Company Services, and Arizona Public Service for funding. We thank Michael Leuthold for running the UA-WRF forecasts. A list of PVLlib-Python contributors may be found on the GitHub repository [17] and in the online documentation [16]. We thank the developers and maintainers of the NOAA forecasts, the Unidata

THREDDS service, and the Unidata Siphon project. We thank Brent Holben, Eric Betterton, John Reagon, and Armin Sorooshian for their effort in establishing and maintaining the Tucson Aeronet site. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and 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.

REFERENCES

- [1] R. W. Andrews, J. S. Stein, C. Hansen, and D. Riley, “Introduction to the open source PV LIB for Python Photovoltaic system modelling package,” in *40th IEEE Photovoltaic Specialist Conference*, 2014.
- [2] W. F. Holmgren, R. W. Andrews, A. T. Lorenzo, and J. S. Stein, “Pvlib python 2015,” in *42nd IEEE Photovoltaic Specialist Conference*, 2015.
- [3] W. F. Holmgren and D. G. Groenendyk, “An open source solar power forecasting tool using pvlib python,” *43rd IEEE Photovoltaic Specialist Conference*, 2016.
- [4] D. P. Larson, L. Nonnenmacher, and C. F. M. Coimbra, “Day-ahead forecasting of solar power output from photovoltaic plants in the american southwest,” *Renewable Energy*, vol. 91, pp. 11–20, 6 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0960148116300398>
- [5] P. A. Jimenez, J. P. Hacker, J. Dudhia, S. Ellen Haupt, J. A. Ruiz-Arias, C. A. Gueymard, G. Thompson, T. Eidhammer, and A. Deng, “Wrf-solar: An augmented nwp model for solar power prediction. model description and clear sky assessment,” *Bulletin of the American Meteorological Society*, 2015/12/14 2015. [Online]. Available: <http://dx.doi.org/10.1175/BAMS-D-14-00279.1>
- [6] W. F. Holmgren. get_nomads. [Online]. Available: https://github.com/wholmgren/get_nomads
- [7] C. Skamarock, J. B. Klemp, J. Dudhia, D. O. Gill, D. M. Barker, M. G. Duda, Xiang-Yu, H. W. Wang, and J. G. Powers, “A description of the advanced research wrf version 3,” NCAR, Tech. Rep. NCAR/TN-475+STR, June 2008.
- [8] M. Leuthold. Arizona regional wrf model forecasts. [Online]. Available: <http://www.atmo.arizona.edu/index.php?section=weather&id=wrf>
- [9] N. C. for Atmospheric Research, “Wrf arw version 3 modeling system user’s guide,” National Center for Atmospheric Research, Tech. Rep., July 2015.
- [10] B. N. Holben, T. F. Eck, I. Slutsker, D. Tanré, J. P. Buis, A. Setzer, E. Vermote, J. A. Reagan, Y. J. Kaufman, T. Nakajima, F. Lavenu, I. Jankowiak, and A. Smirnov, “Aeronet—a federated instrument network and data archive for aerosol characterization,” *Remote Sensing of Environment*, vol. 66, no. 1, pp. 1–16, 1998. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0034425798000315>
- [11] R. E. Bird and R. L. Hulstrom, “Direct insolation models,” Solar Energy Research Institute, Tech. Rep. SERI/TR-33S-344, 1981.
- [12] A. P. Dobos, “Pvwatts version 5 manual,” National Renewable Energy Laboratory, Tech. Rep., 2014.
- [13] P. Loutzenhiser, H. Manz, C. Felsmann, P. Strachan, T. Frank, and G. Maxwell, “Empirical validation of models to compute solar irradiance on inclined surfaces for building energy simulation,” *Solar Energy*, vol. 81, no. 2, pp. 254 – 267, 2007. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0038092X06000879>
- [14] W. De Soto, S. A. Klein, and W. A. Beckman, “Improvement and validation of a model for photovoltaic array performance,” *Solar Energy*, vol. 80, no. 1, pp. 78–88, 1 2006. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0038092X05002410>
- [15] D. King, W. Boyson, and J. Kratochvill, “Photovoltaic array performance model,” Sandia National Laboratories, Tech. Rep. SAND2004-3535, 2004.
- [16] PVLlib Python contributors. pvlib-python documentation. [Online]. Available: <https://pvlib-python.readthedocs.io>
- [17] ———. pvlib/pvlib-python. [Online]. Available: <https://github.com/pvlib/pvlib-python>
- [18] A. Andreas and S. Wilcox, “Observed atmospheric and solar information system (oasis); tucson, arizona (data),” NREL, Tech. Rep. NREL Report No. DA-5500-56494, 2010.