A Comparison of PV Power Forecasts Using PVLib-Python

William F. Holmgren[1], Antonio T. Lorenzo[1], and Clifford Hansen[2]
[1]Dept. of Hydrology and Atmospheric Sciences, Univ. of Arizona, Tucson, AZ
[2]Sandia National Laboratories, Albuquerque, NM

Introduction

PVLib-Python is an open source toolbox for PV modeling. PVLib was originally developed at Sandia National Laboratories and has been expanded by contributions from members of the Photovoltaic Performance and Modeling Collaboration (PVPMC). The PVLib source code is hosted on GitHub. PVLib-Python and PVLib MATLAB are BSD 3-clause licensed. We encourage users to contribute the library at github.com/pvlib and ask questions on stackoverflow using the pvlib tag or on the mailing list.

In this paper, we use the PVLib-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. As an example of the utility of PVLib-Python for creating benchmark forecasts, we compare the forecasts derived from NOAA weather models (GFS, NAM, and RAP) with forecasts derived from a model run by U. Arizona.

Irradiance Forecast Data

The most critical component of a PV power forecast is the forecast of 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. 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. The table below shows irradiance and cloud cover model field data availability for the studied models. For NOAA models, this data only reflects availability on the NOMADS THREDDS server.

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

For the NOAA models studied here, we use a model proposed by Larson et. al. to calculate GHI from cloud cover forecasts:

```
ghi = (offset + (1 - offset) * (1 - cloud_cover)) * ghi_clear
```

where `offset=0.35`, `cloud_cover` is the total cloud cover, and `ghi_clear` is determined by PVLib's climatological clear sky model. The DISC model is then used to calculate DNI and DHI.

We post-processed the UA-WRF model's irradiance forecasts using measurements of the previous day's average aerosol optical depth obtained from the Aeronet site in Tucson, AZ, according to the equations below.

$$\mathrm{DNI} = \mathrm{DNI}_{\mathrm{wrf}} \exp(-\tau_{bb}/\cos\theta_z)$$
$$\mathrm{GHI} = \mathrm{GHI}_{\mathrm{wrf}} \exp(-0.01/\cos^{0.4}\theta_z)$$

The table below summarizes the combinations of weather model data and processing algorithms studied in this paper.

Name	GHI	DNI
GFS-CC	NAM 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

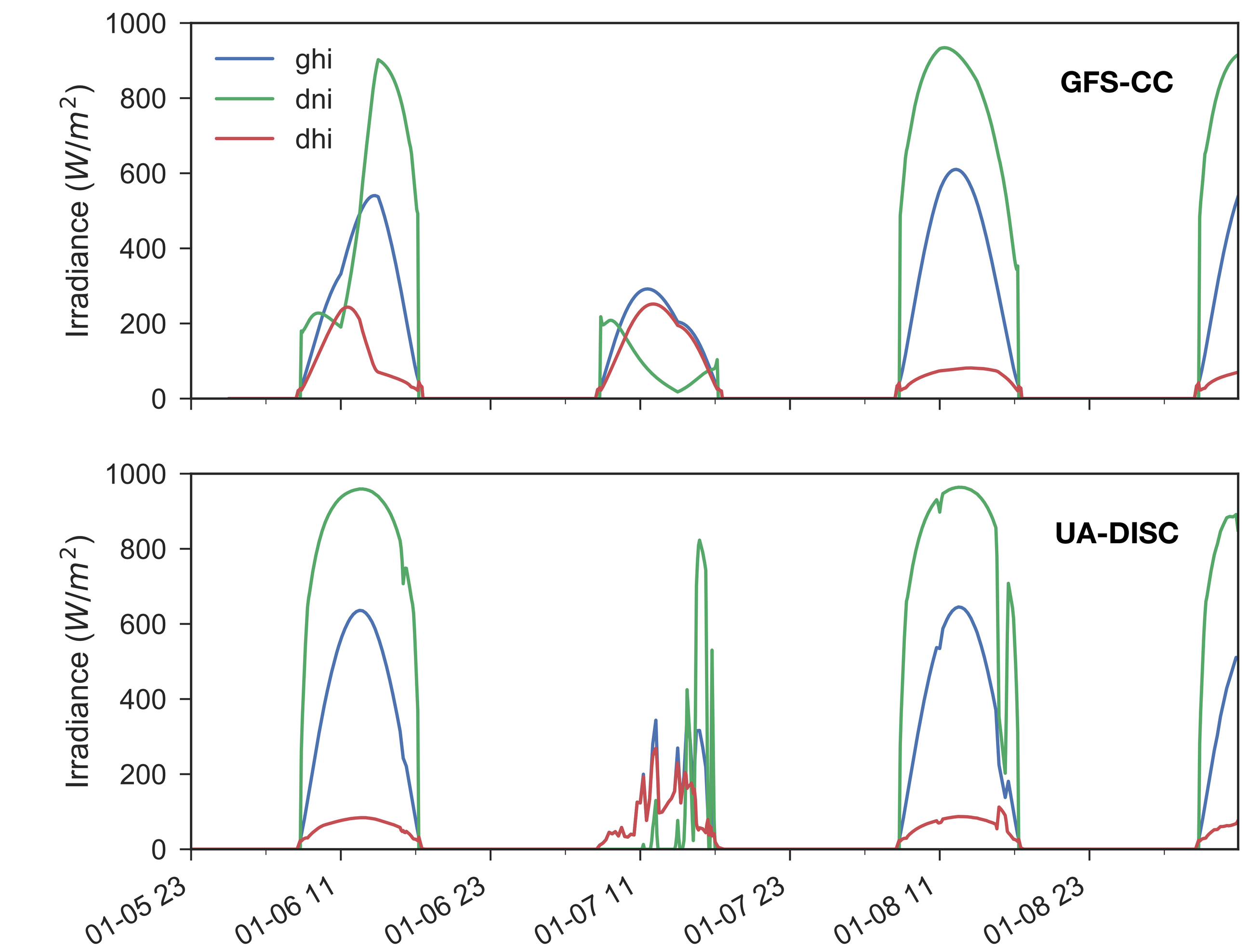


Fig. 1. Irradiance derived from 3 hourly GFS cloud cover (top) and UA (bottom) models.

We linearly interpolate the model forecast data from its native resolution to 5 minute resolution. For the GFS, NAM, and RAP models we use the Larson and the DISC model to determine a forecast GHI, DNI, and DHI. For the NAM model, we also create forecasts directly from its GHI forecasts.

Forecasting PV Power

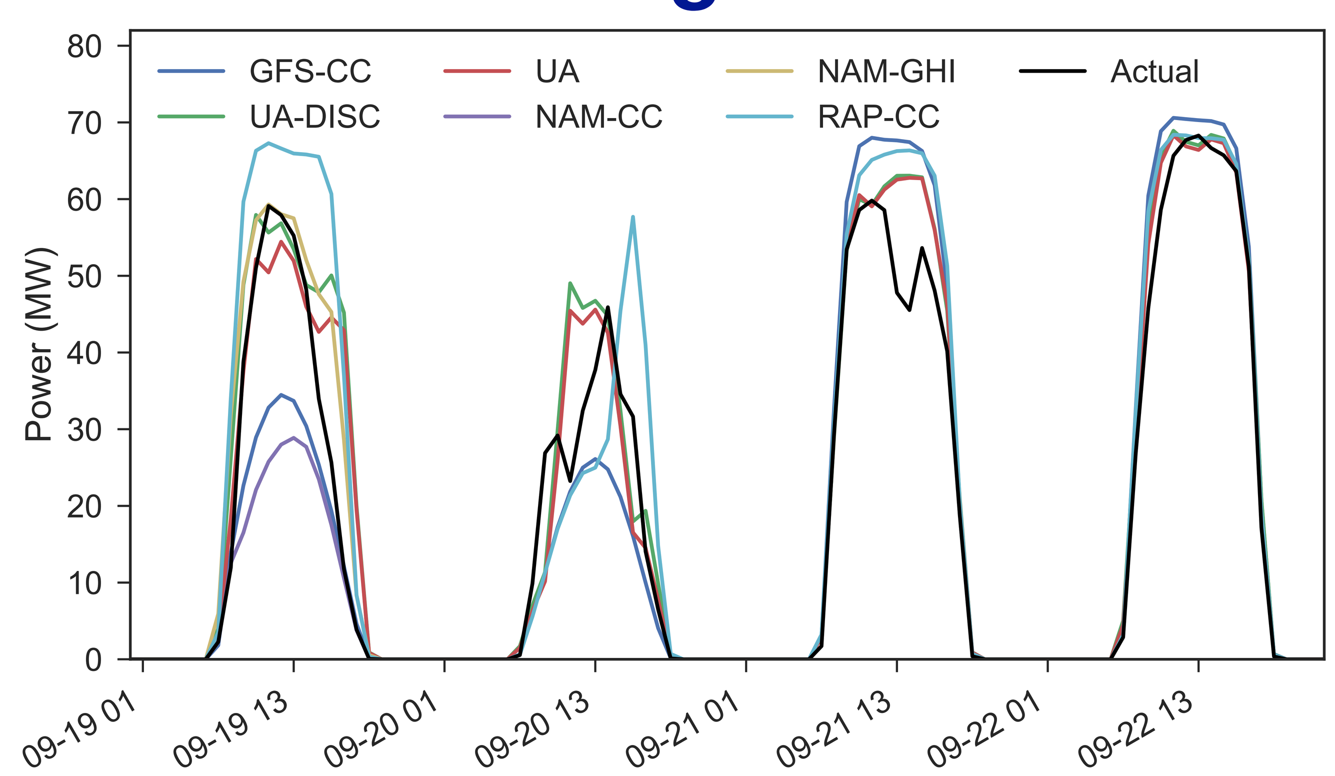


Fig. 2. Four days of PV generation (black) and forecasts (colors) derived from the GFS, NAM, RAP, and UA-WRF models using cloud cover (CC) or irradiance forecasts.

We created forecasts for six PV systems in Arizona. The systems included three single axis trackers (63 MW AC), and three fixed tilt systems (14 MW AC). Five of the six systems are located near Tucson, Arizona. Aggregate forecasts are shown in Fig. 2.

To model PV generation for each system, we used PVWatts with a DC nameplate capacity and temperature coefficient. We also imposed a maximum AC capacity parameter to account for inverter clipping. We determined system parameters by manually optimizing forecast model performance for clear days.

```
location = Location(latitude=32.2, longitude=-110.9, altitude=700)
system = SingleAxisTracker(
    module_parameters={'pdc0': 10.0, 'gamma_pdc': -0.0035})
system.peak_ac_power = 9.0
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()
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()
```

Results

We compared the accuracy of all of the NOAA forecast models as a function of the forecast horizon. Times at which any forecast was missing were removed from the comparative analyses.

Fig. 3. NOAA models Aug–Dec 2016

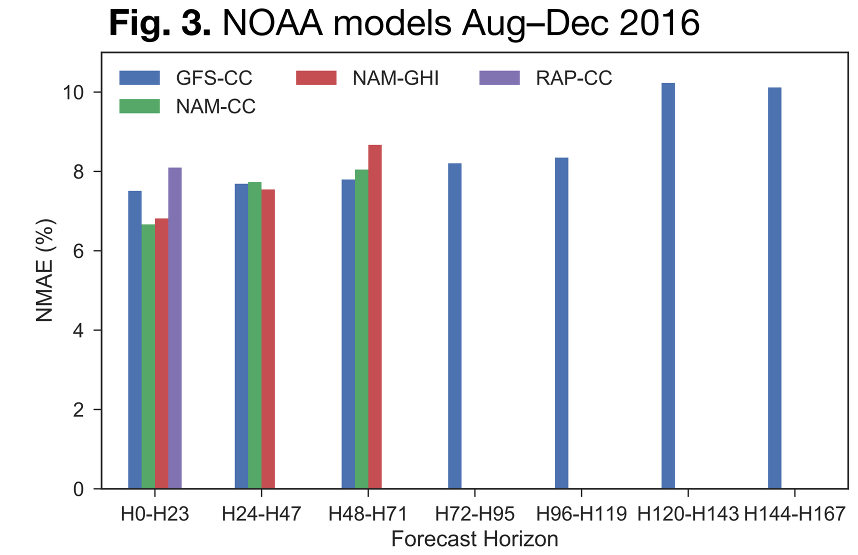
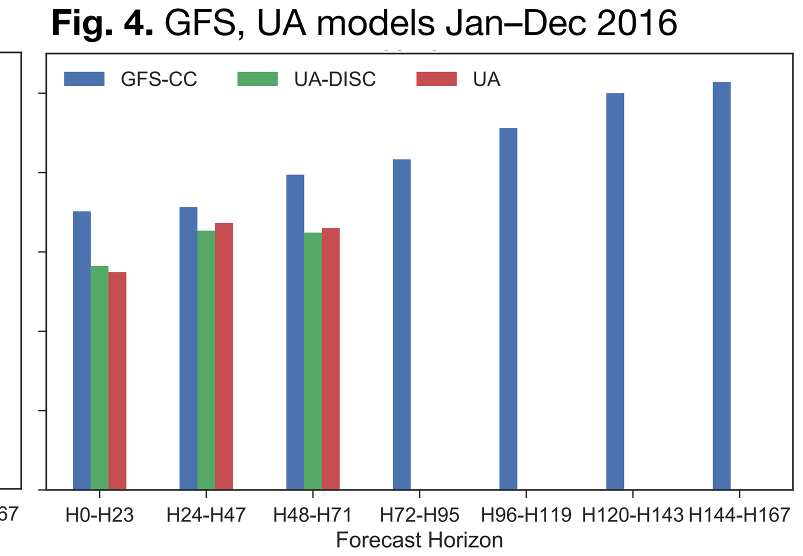
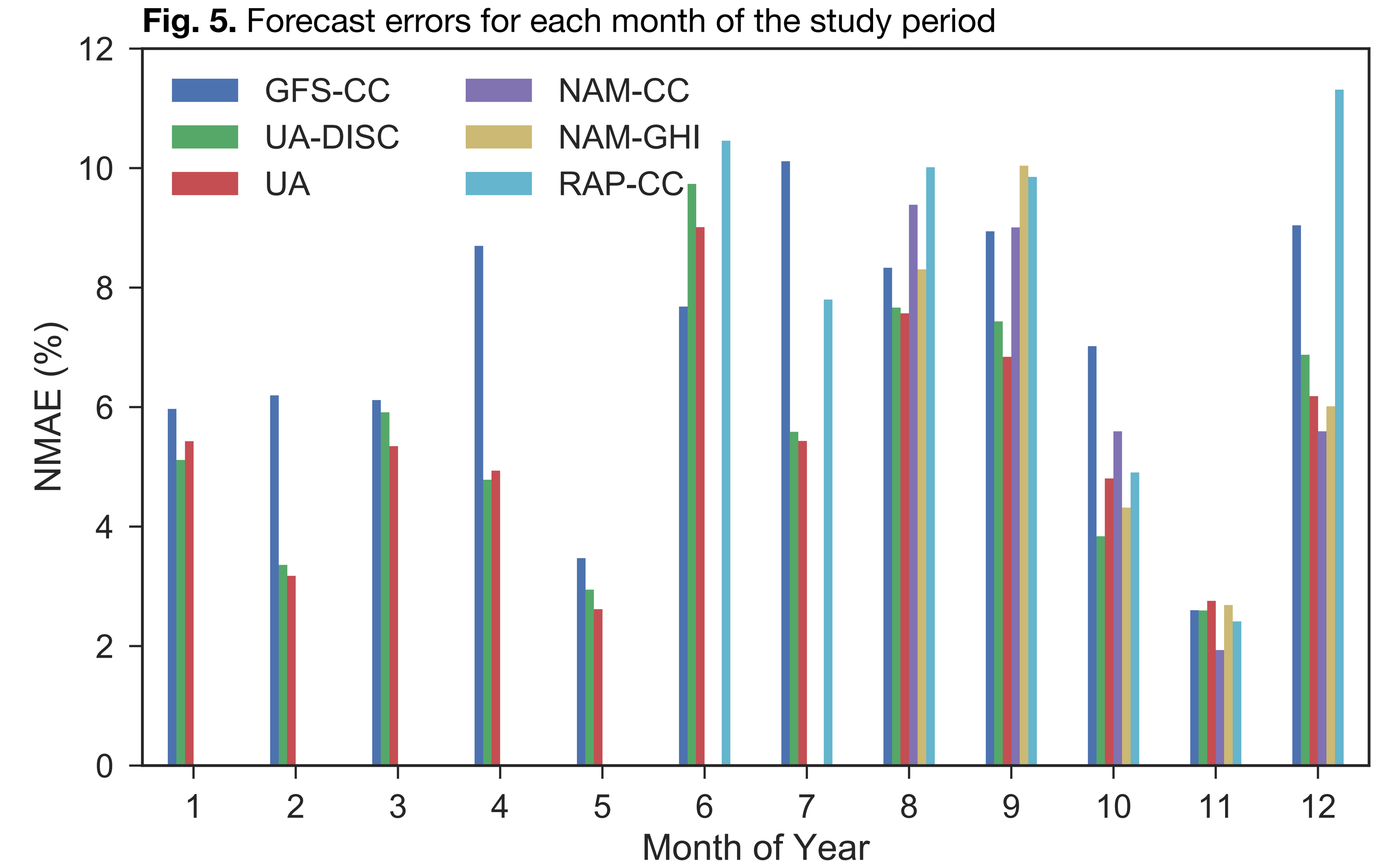


Fig. 4. GFS, UA models Jan–Dec 2016



Next, we examined forecast accuracy as a function of month of year. Figure 5 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.

Fig. 5. Forecast errors for each month of the study period



These results led us to examine the relationship between forecast accuracy and clear sky condition. We used PVLib's `detect_clear` function to determine if a minute is clear or not, summed the number of cloudy minutes in each month, and normalized by the number of daylight minutes. Relatively clear months have lower errors (Fig 6).

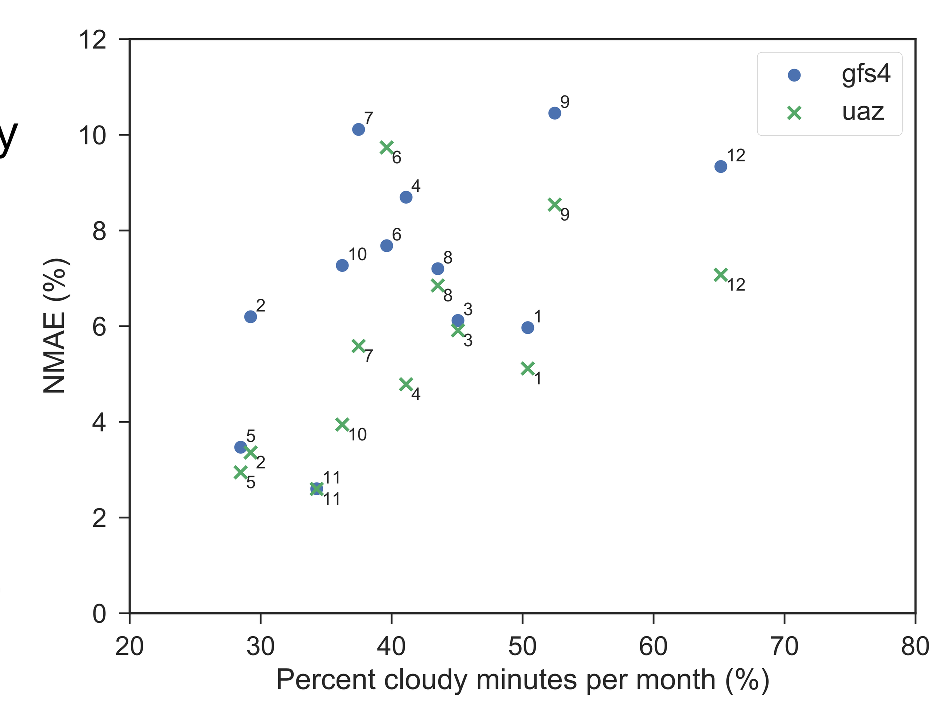


Fig. 6. Forecast errors vs. % cloudy minutes/month

Acknowledgements

We gratefully acknowledge the Electric Power Research Institute, Southern Company Services, Tucson Electric Power, and Arizona Public Service for support. We thank Michael Leuthold for running the UA-WRF forecasts. A list of PVLib-Python contributors may be found online. 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.

https://github.com/pvlib/pvlib-python