



Model Validation- The Motivation for the Session

Janine Keith
2025 PVPMC
May 13, 2025

Photo by Josh Bauer / Bryan Bechtold, NREL 82067

Why do we validate models?

A model is always going to be... a little bit wrong

- A lot of inputs are hard to quantify
- There's uncertainty, even in the **measured** inputs
- Some simplifications are required for feasible models
- And, even with great models... there is user “variation”

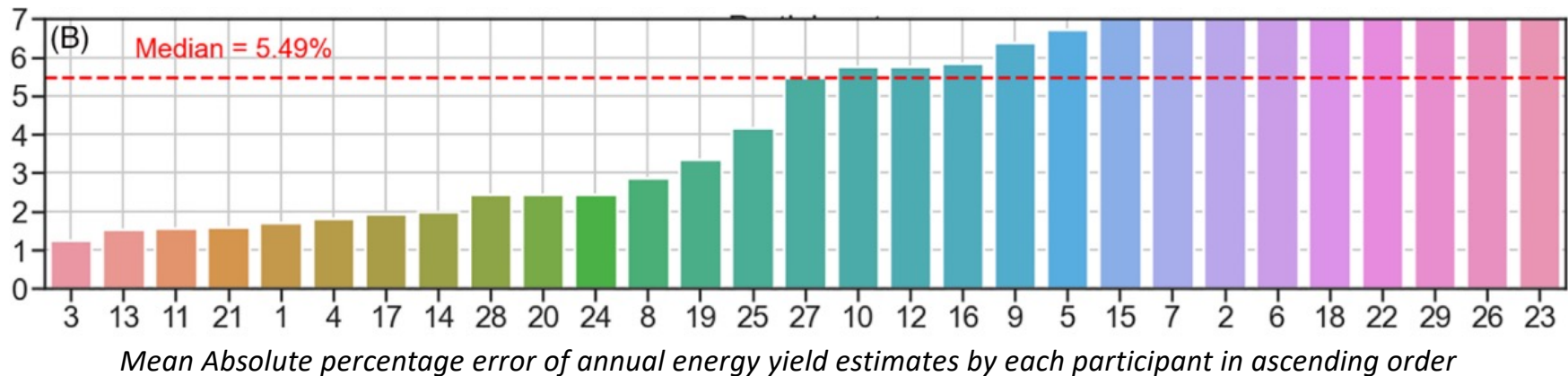


Figure 4 from: Theristis M, Riedel-Lyngskær N, Stein JS, et al. Blind photovoltaic modeling intercomparison: A multidimensional data analysis and lessons learned. Prog Photovolt Res Appl. 023;31(11):1144-1157. doi:10.1002/pip.3729

But... financial decisions are made based on model results.

So we need to know that a model is reasonably accurate.

How do we do that?

Option 1: Methodology review

Detailed manual review of the PV modeling algorithm implementation by a knowledgeable source outside of the core programming team

```
void exec()
{
    std::unique_ptr<weather_data_provider> wdprov;

    if (is_assigned("solar_resource_file"))
    {
        const char* file = as_string("solar_resource_file");
        wdprov = std::unique_ptr<weather_data_provider>(new weatherfile(file));

        weatherfile* wfile = dynamic_cast<weatherfile*>(wdprov.get());
        if (!wfile->ok()) throw exec_error("pvwattsV8", wfile->message());
        if (wfile->has_message()) log(wfile->message(), SSC_WARNING);
    }
    else if (is_assigned("solar_resource_data"))
    {
        wdprov = std::unique_ptr<weather_data_provider>(new weatherdata(lookup("solar_resource_data")));
        if (!wdprov->ok()) {
            throw exec_error("pvwattsV8", wdprov->message());
        }
        if (wdprov->has_message()) log(wdprov->message(), SSC_WARNING);
    }
    else
        throw exec_error("pvwattsV8", "No weather data supplied.");

    pv.dc_nameplate = as_double("system_capacity") * 1000; //units of this variable are W, while input is in kW
    pv.dc_ac_ratio = as_double("dc_ac_ratio");
    pv.ac_nameplate = pv.dc_nameplate / pv.dc_ac_ratio;

    pv.xfmr_rating = pv.ac_nameplate;

    pv.inv_eff_percent = as_double("inv_eff");

    size_t soiling_len = 0;
    ssc_number_t* soiling = nullptr;
    if (is_assigned("soiling"))
    {
        soiling = as_array("soiling", &soiling_len);
    }
}
```

Image: screenshot of PVWattsV8 code

Option 2: Compare a tool to another tool

Also called “benchmarking”, this approach can help assure users who are already comfortable with the existing tool that the new tool gives similar results.

New model : Reality $\approx$ Existing model : Reality

“Shading validation” studies compare a tool’s prediction of annual solar access value to what a SunEye would predict



Option 3: Compare a tool to measured data

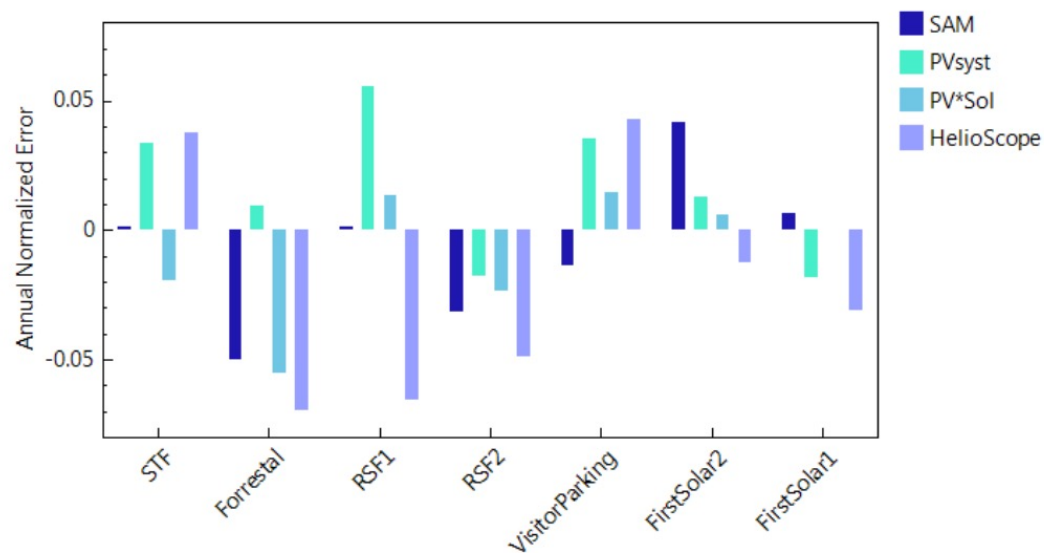


Figure 1. Annual normalized error for each tool by system

No data for PV*SOL for FirstSolar1.

Figure from Guittet, Darice and Janine Freeman. 2018. Validation of Photovoltaic Modeling Tool HelioScope Against Measured Data. Golden, CO: National Renewable Energy Laboratory. NREL/TP-6A20-72155. <https://docs.nrel.gov/docs/fy19osti/72155.pdf>.

Comparing to
measured data
seems like it would
be the gold
standard!

... but it turns out that comparing to measured data is messy because data can be wrong too

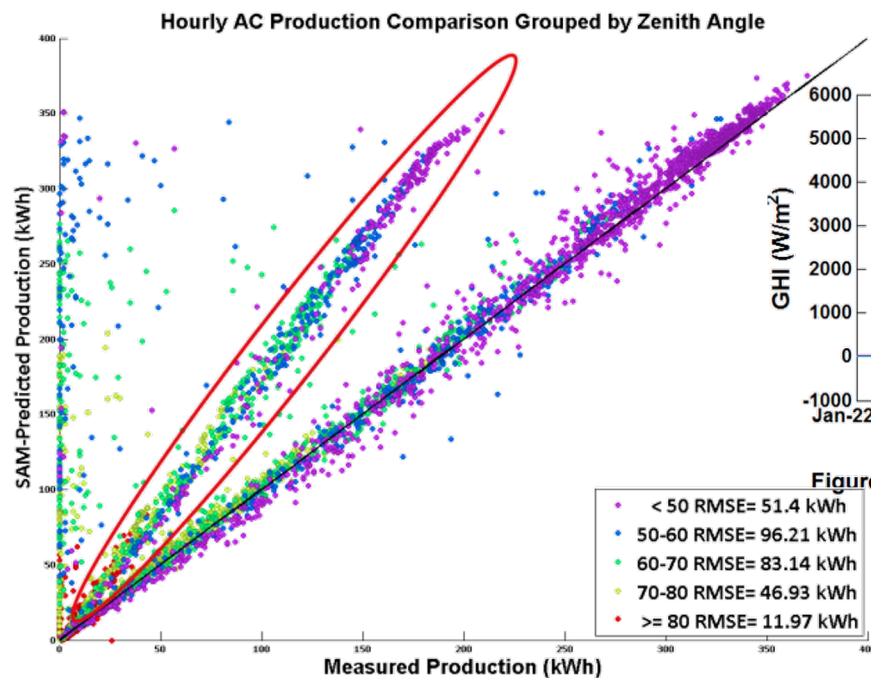


Figure 1-4. Second trendline formed by inverter outages

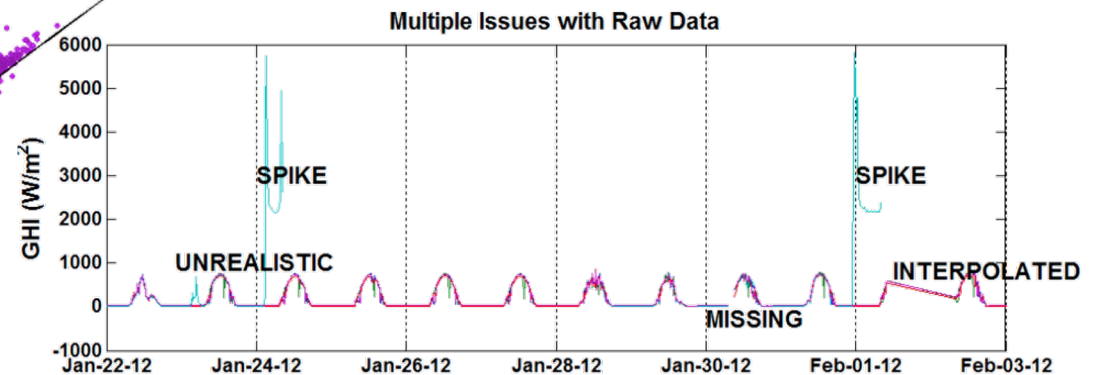


Figure 1-2 Examples of multiple data quality issues present in redundant GHI data streams

Figures from Freeman, Janine, Jonathan Whitmore, Leah Kaffine, et al. 2013. System Advisor Model: Flat Plate Photovoltaic Performance Modeling Validation Report. Golden, CO: National Renewable Energy Laboratory. NREL/TP-6A20-60204. <https://docs.nrel.gov/docs/fy14osti/60204.pdf>.

So why are we having this session?

The Challenge

- There continue to be a steady stream of new entrants to the PV modeling tool space.
- Different tools have different advantages.
 - *Accuracy*
 - *Speed*
 - *Design features*
 - *Automation*
 - *Suitability for large analyses*

We need to be able to use the best tool for the job at hand and trust that the results are reasonable for our purposes.

“Bankability” is a nebulous target. We need a better process.

The Session

- 1** Lelia Deville- A research perspective
- 2** Imre Horvath- A modeling tool perspective
- 3** Jeff Newmiller- An independent engineering perspective
- 4** Moderated discussion

How Do We “Validate” Models More Efficiently?

- How do we know when a tool is “good enough”?
Is this different for residential/commercial vs utility scale?
- Who needs to validate a tool?
Is there an acceptable way for a company to do it themselves?
- How do we value tradeoffs?
Speed and design features vs accuracy and detailed component modeling
- How should **we** define “bankable”?
- How can we convince financiers of this definition?