Implementing Machine Learning in the PCWG Tool

Andrew Clifton, NREL

Yu Ding, Texas A&M University

Peter Stuart, RES

Power Curve Working Group Meeting
Glasgow, Scotland
December 13, 2016

NREL/PR-5D00-67641

REVISED FOR PUBLICATION DECEMBER 21, 2016

What are we trying to do?

Goals

- Want to apply machine learning in the context of the Power Curve Working Group experience sharing exercises
- Want to predict turbine performance in the inner and outer ranges
- Want to find out what type of machine learning methods are the most capable.
- Want to make a solution transparent to the end user

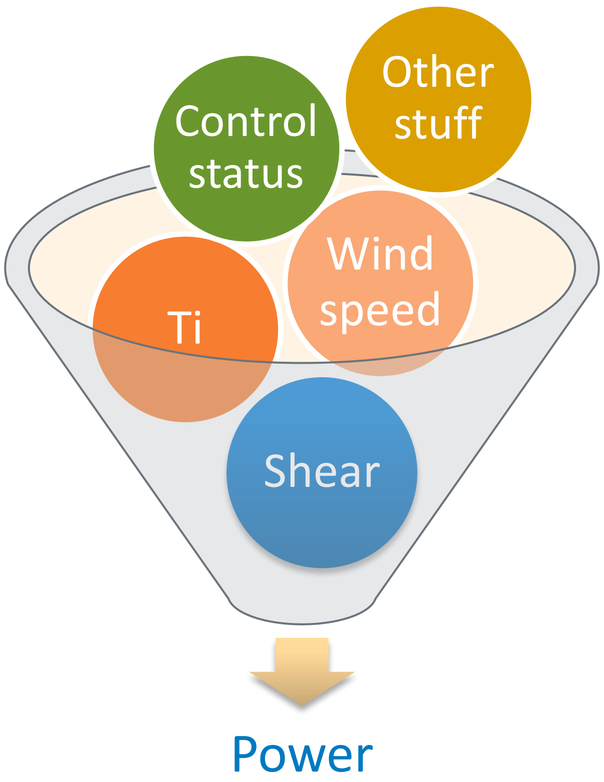
Challenges

- Machine learning works best with lots of training data that covers all of the operating envelope…
- … but we need to maintain participant confidentiality.

See www.pcwg.org for more information about the Power Curve Working Group

What is Machine Learning?

The concept

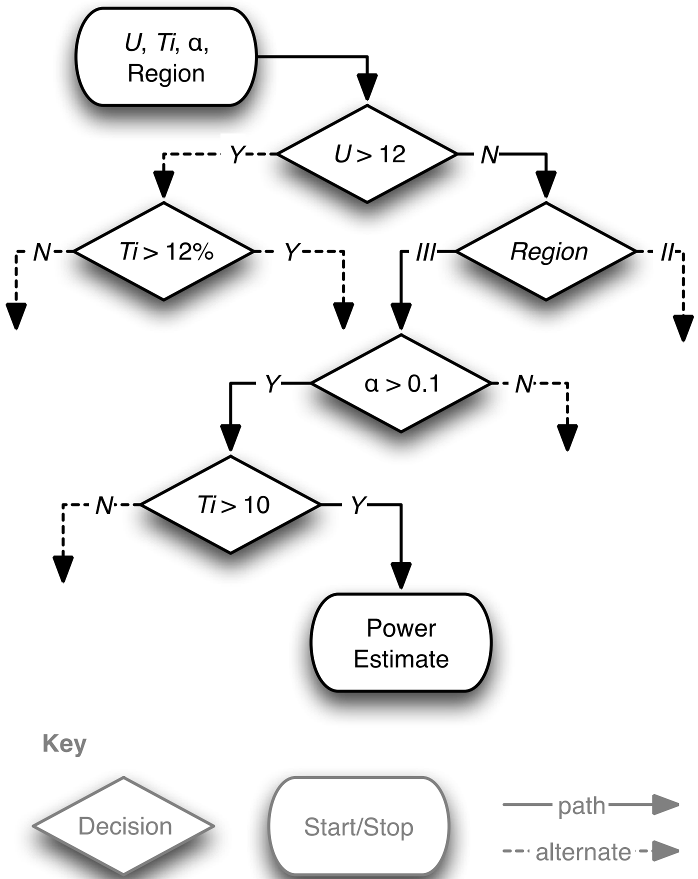


Some details

- Train model from experience
- Create a transfer function from inflow to power
- Use whatever inputs are available
  - Wind speed
  - Rotor-disk shear
  - Turbulence intensity
  - Turbine controller state
  - …
- Just need one output
  - 10-minute power, or
  - 10-minute difference in power v. OEM power curve

Example model using “random forests”

- Train the model with 1000s of observations
  - Create a tree, with lots of branches
  - Estimate 10-minute power
- Use 100 trees in an ensemble
  - Output is the ensemble mean
  - Get ensemble standard deviation as well
- 2 or 3 lines of code



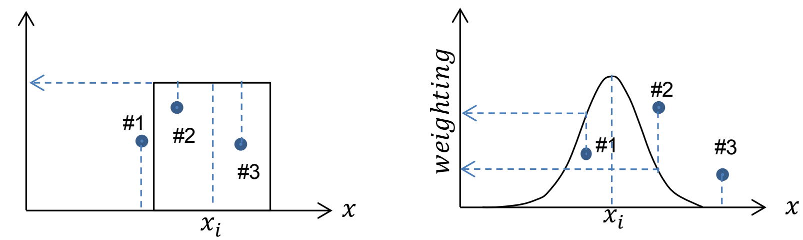
One branch of a regression tree in a Random Forest.

Source: Clifton, A., L. Kilcher, J.K. Lundquist, and P. Fleming. 2013. “Using machine learning to predict wind turbine power output.” Environmental Research Letters 8 (2)

Example model using "kernel method"

Kernel PLUS versus IEC binning

- Kernel PLUS uses a smooth window



(a) Binning: equal weighting **(b) Kernel: different weighting**

- Kernel PLUS scales up to multi-dimension inputs:

$$Power\ prediction \leftarrow kernel(V, D, density) + kernel(V, D, humidity) + kernel(V, D, TI) + kernel(V, D, Wind\ shear) + (others\ when\ available) \dots$$

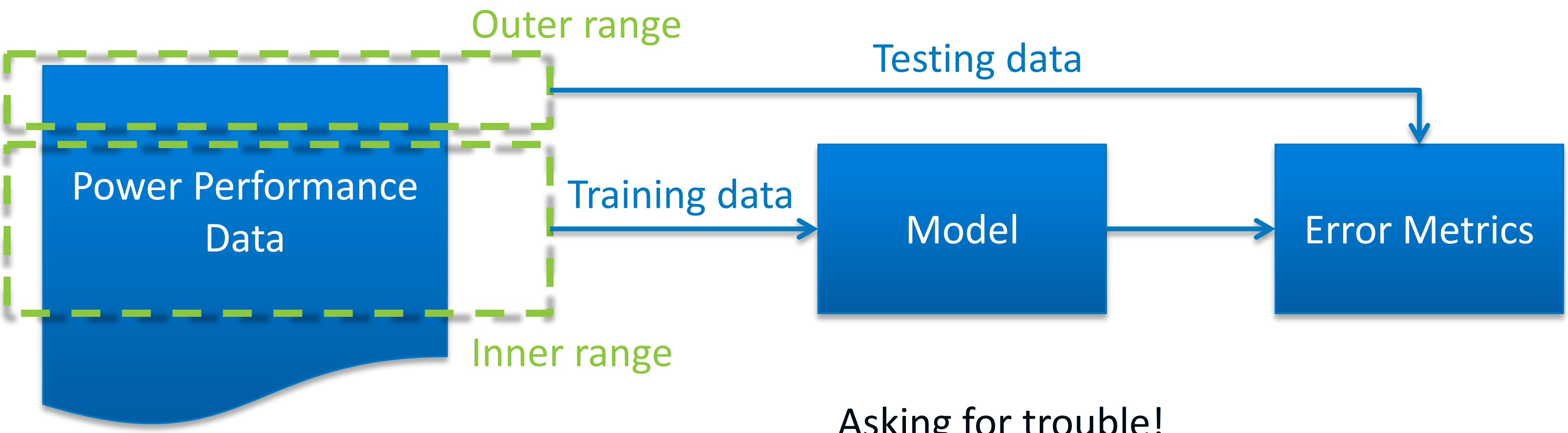
Source: Lee, Ding, Genton, Xie. 2015. "Power curve estimation with multivariate environmental factors for inland and offshore wind farms." *Journal of the American Statistical Association*, 110: 56-67

Model elements

- Current version uses up to seven inputs
  - Wind speed (V)
  - Wind direction (D)
  - Air density (ρ)
  - Humidity (H)
  - TI (S)
  - Above hub wind shear (W_a)
  - Below hub wind shear (W_b)
- The model works with a subset of the inputs and it performs similarly to IEC binning if only wind speed is used.
- Model trained with 80% of a one-year worth of ten-minute data.

Some Options

1. Using inner range data to train the model

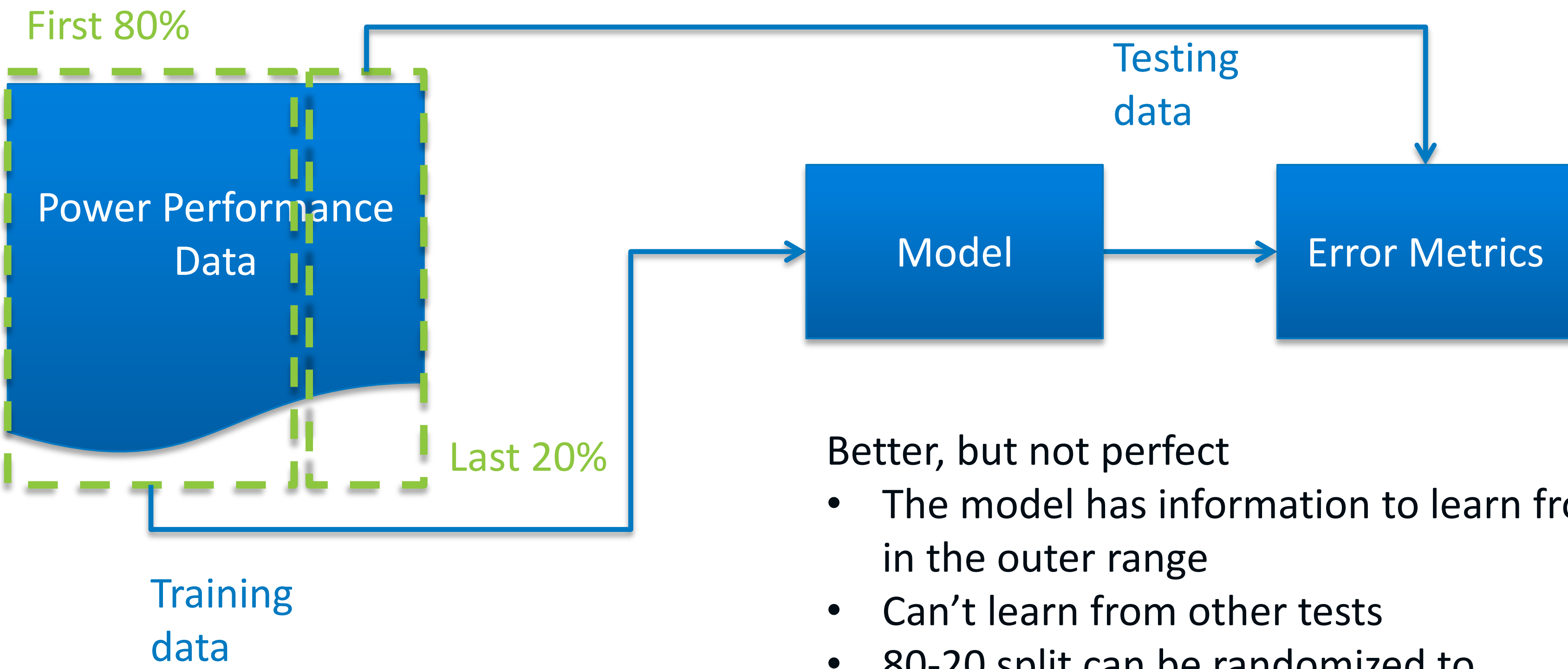


Asking for trouble!

- The model doesn't have any information to learn from in the outer range
- Only going to look bad for machine learning…

Some Options

2. Striping within one data set

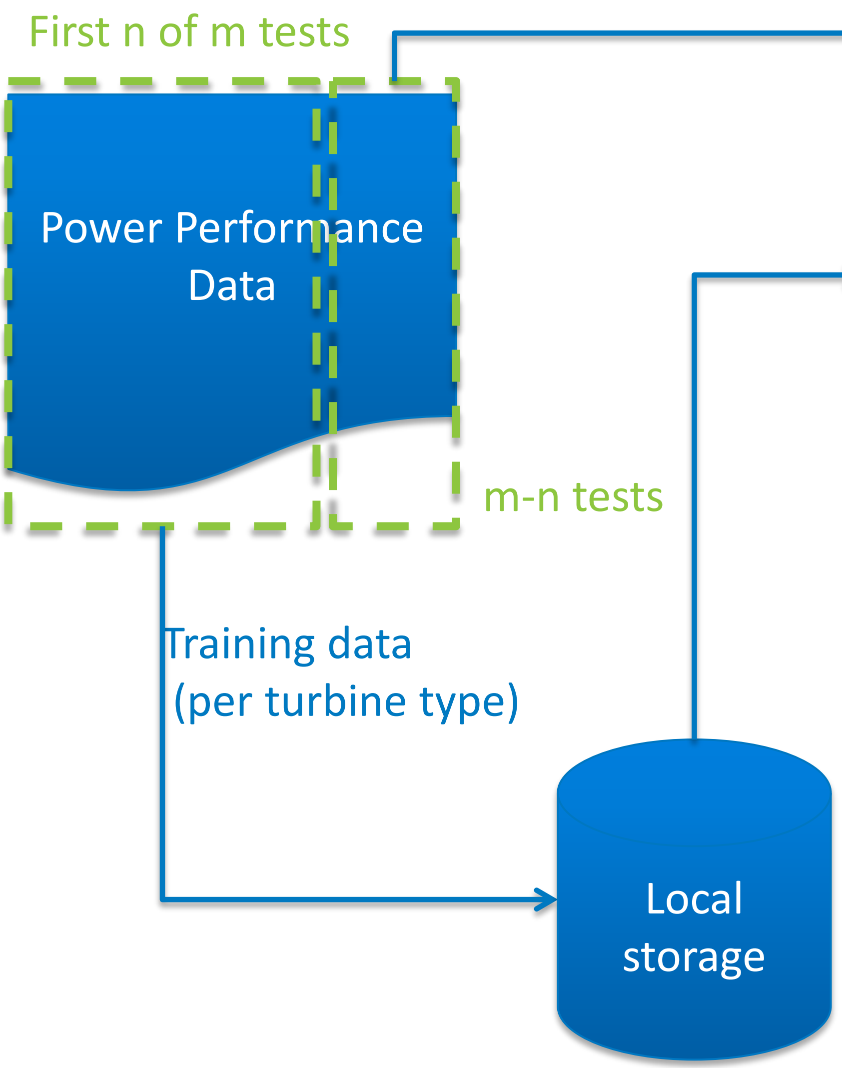


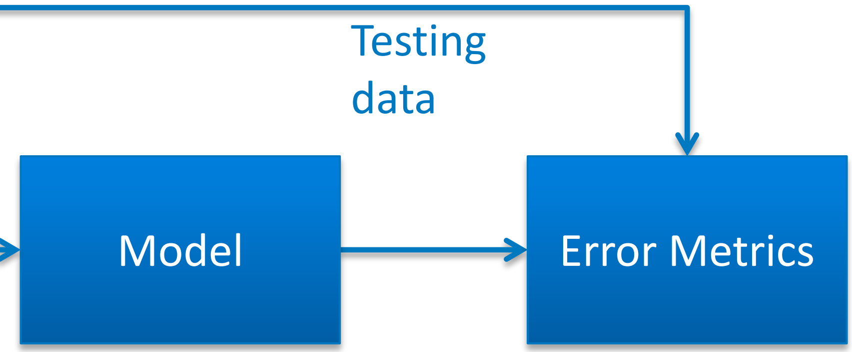
Better, but not perfect

- The model has information to learn from in the outer range
- Can’t learn from other tests
- 80-20 split can be randomized to compute a 5-fold cross validation error metric.

Some Options

3. Local training corpus



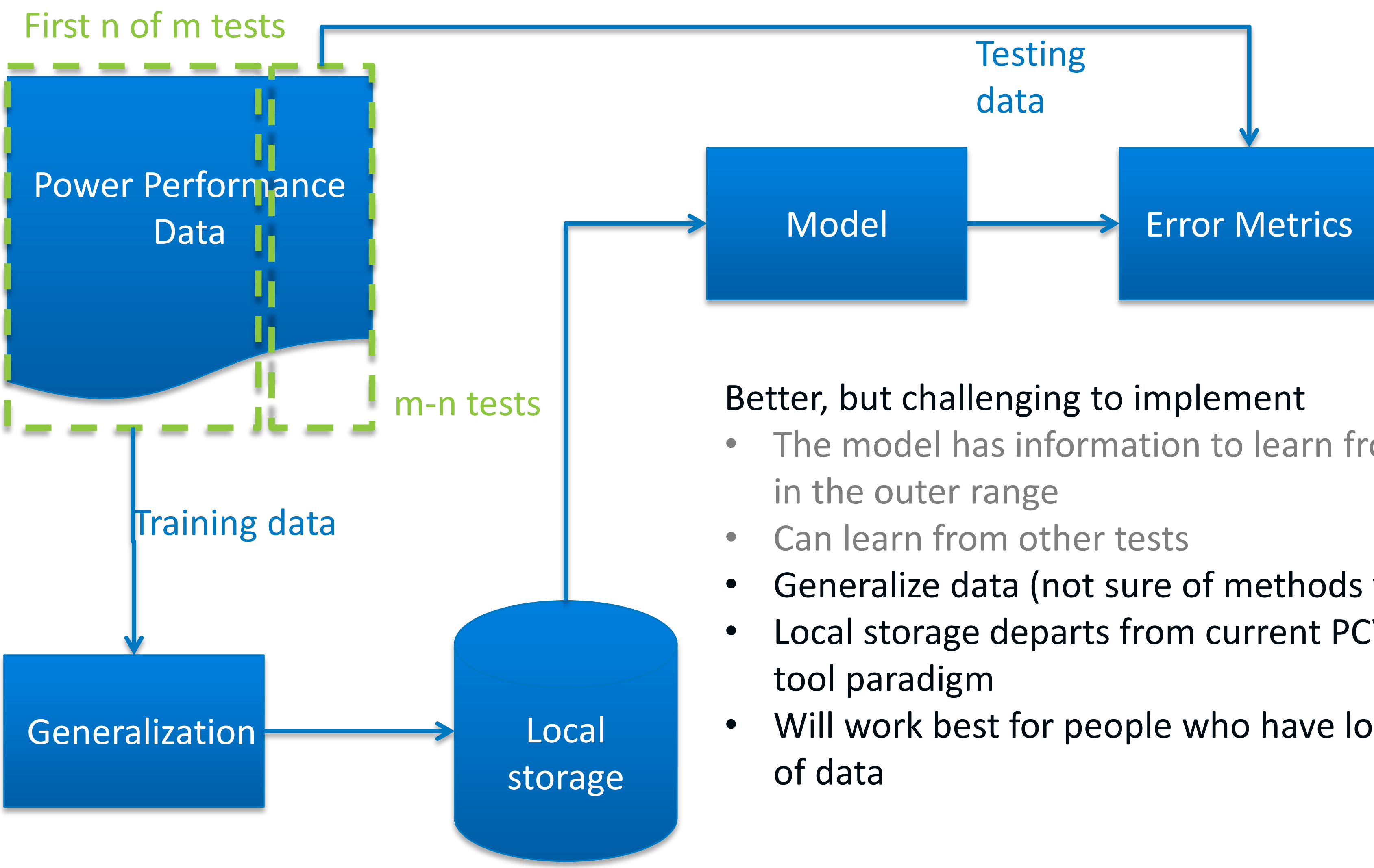


Better, but challenging to implement

- The model has information to learn from in the outer range
- Can learn from other tests
- Collect data from individual turbine types
- Local storage departs from current PCWG tool paradigm

Some Options

4. Local generalized training corpus

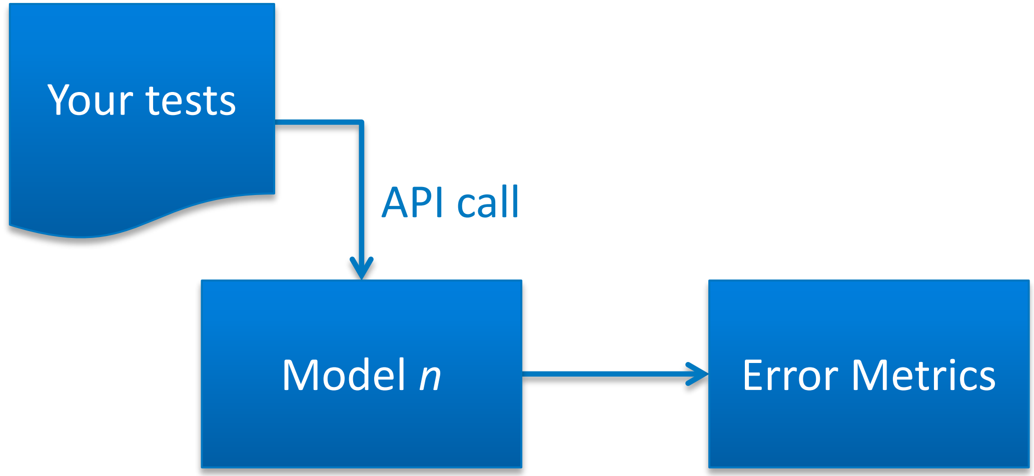


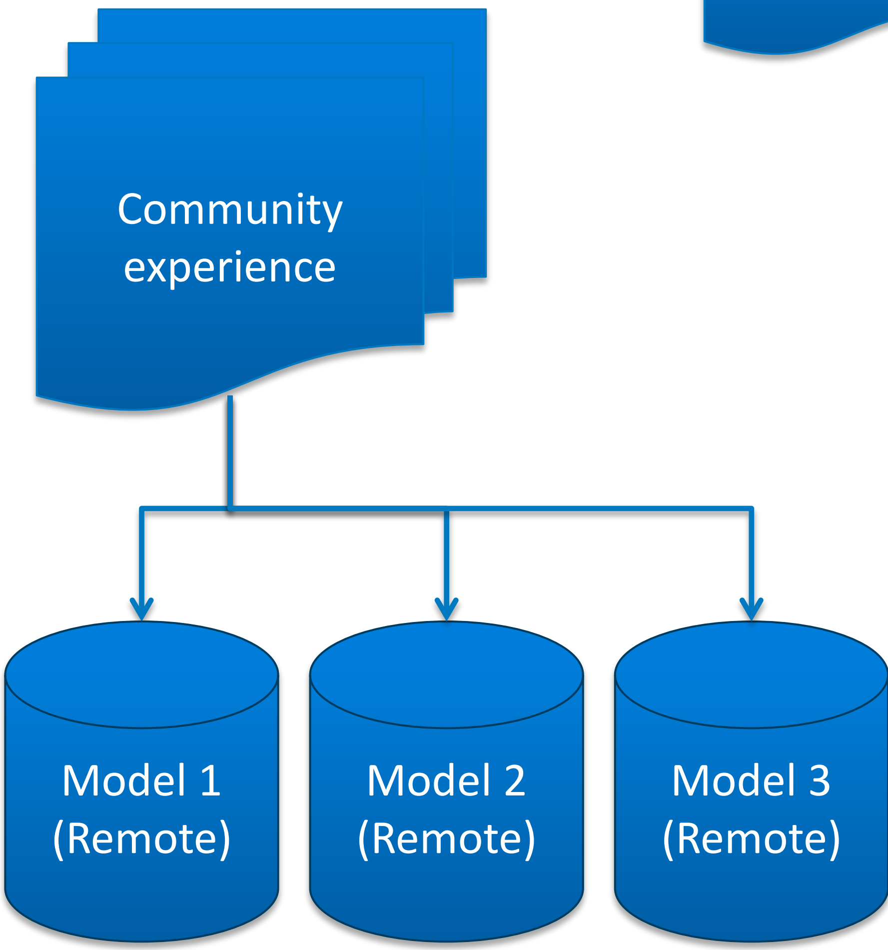
Better, but challenging to implement

- The model has information to learn from in the outer range
- Can learn from other tests
- Generalize data (not sure of methods yet)
- Local storage departs from current PCWG tool paradigm
- Will work best for people who have lots of data

Some Options

5. Common generalized training corpus, multiple remote models





Better, but challenging to implement

- The model has information to learn from in the outer range
- Can learn from other tests
- Generalize data (not sure of methods yet)
- Anyone can contribute a performance model
- Anyone can test any model if they know the URL

Machine Learning in PCWG-Share-X

- Two normalised training datasets have been generated using 6 actual power performance test datasets.
- **These datasets will be made available to any PCWG member who will commit to providing a machine learning implementation for PCWG-Share-X (email pcwg@res-ltd.com for access).**

The datasets have been normalised as follows:

$$\text{Normalised Wind Speed} = \frac{\text{Air Density Normalised Wind Speed} - \text{Cut In Wind Speed}}{\text{Rated Wind Speed} - \text{Cut In Wind Speed}}$$

Where cut-in wind speed and rated wind speed are defined from the zero TI power curve.

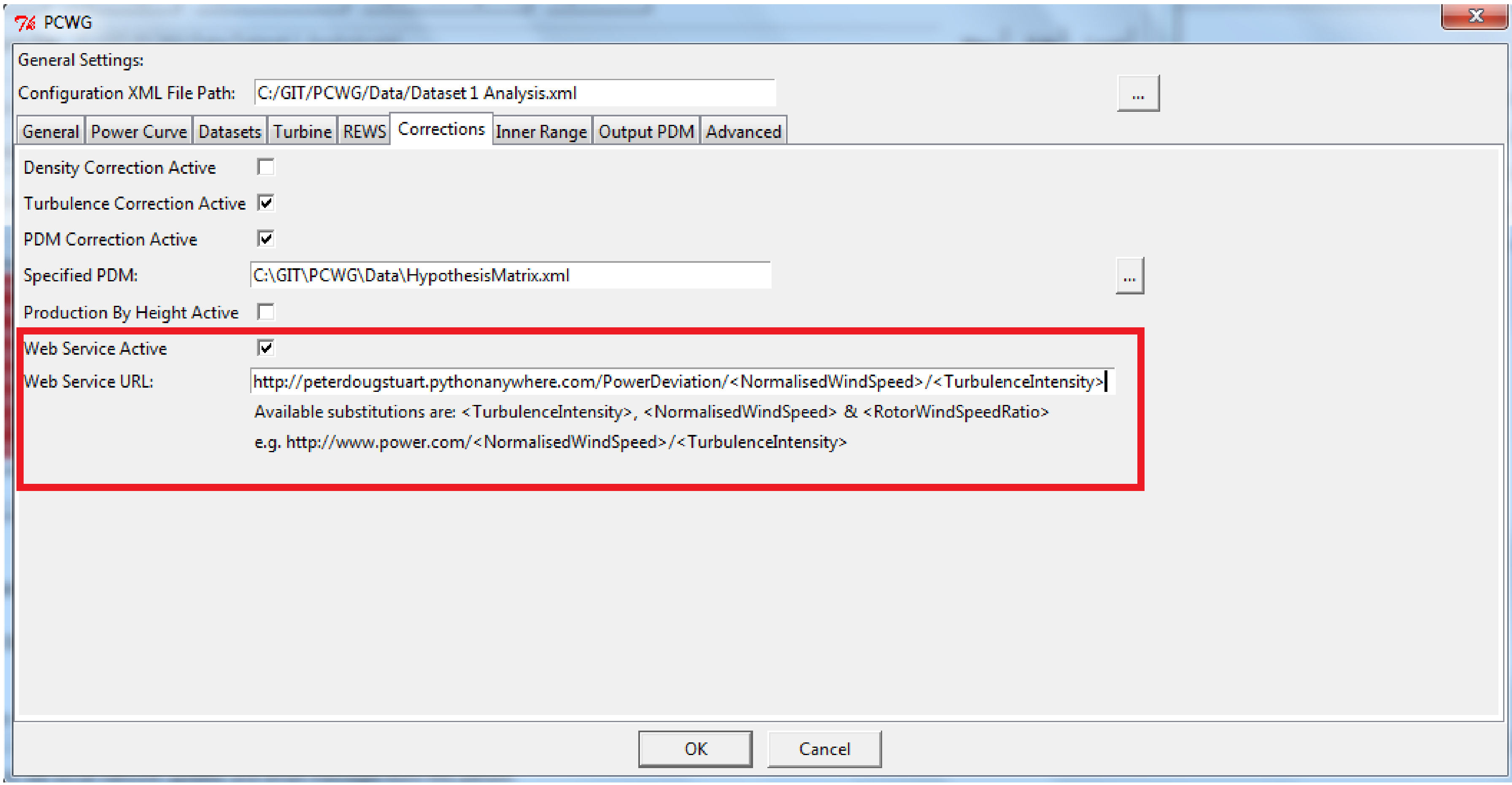
Power Deviation = (Actual - Predicted) / Predicted

Rotor Wind Speed Ratio $RSWR = U(z_H + \frac{3}{4}R) / U(z_H - \frac{3}{4}R)$

The year in the time stamp is a randomly generated value (this preserves seasonality and diurnal information).

Machine Learning in PCWG-Share-X

- A feature has been added to the PCWG Tool to integrate it with a machine learning implementation running on a remote server
- As an example, Peter set up a simple web service with a 'fake' implementation here: http://peterdougstuart.pythonanywhere.com/PowerDeviation/0.8/0.12.



Let’s talk!

andrew.clifton@nrel.gov
+1 303 513 0095

A multi-megawatt wind turbine and 1 MW PV field at NREL's National Wind Technology Center
Golden, CO. Image by Dennis Schroeder, NREL