Forecasting Wind Power

Luc Demortier

Tuesday, 1 August 2017

A hackathon...

On July 19 and 20, 2016, the H2O Open Tour came to the IAC building in New York City to present their product and foster community with the help of tutorials, talks, social events, and a hackathon.

After H2O's event I got my hands on the full hackathon data set (including all the answers!), and kept working on it for a while...

The hackathon's challenge

...was to predict the power output of an array of ten wind turbines arranged in ten "zones":

- Each zone contains one turbine and one wind speed measuring station.
- Each wind speed measuring station delivers four measurements: zonal and meridional wind velocities at heights of 10 and 100 meters above ground.

The data set: contains 168,000 hourly measurements made over a period of 100 weeks, from January 1, 2012 to December 1, 2013:

Data Set	Data-Taking Period	Number of observations
Training	01/01/2012 - 07/31/2013	138,710
Public Leaderboard	08/01/2013 - 09/30/2013	14,640
Private Leaderboard	10/01/2013 - 12/01/2013	14,650

The data

Each "row" in the data file contains the following fields:

Feature	Description
ID	Measurement ID
ZONEID	Zone id: an integer between 1 and 10
TIMESTAMP	Date and time of measurement
TARGETVAR	Turbine power output, a number between 0 and 1
U10	Zonal wind velocity 10 meters above ground
V10	Meridional wind velocity 10 meters above ground
U100	Zonal wind velocity 100 meters above ground
V100	Meridional wind velocity 100 meters above ground

(Each turbine's power output is normalized to the turbine's maximal capacity.)

The performance measure

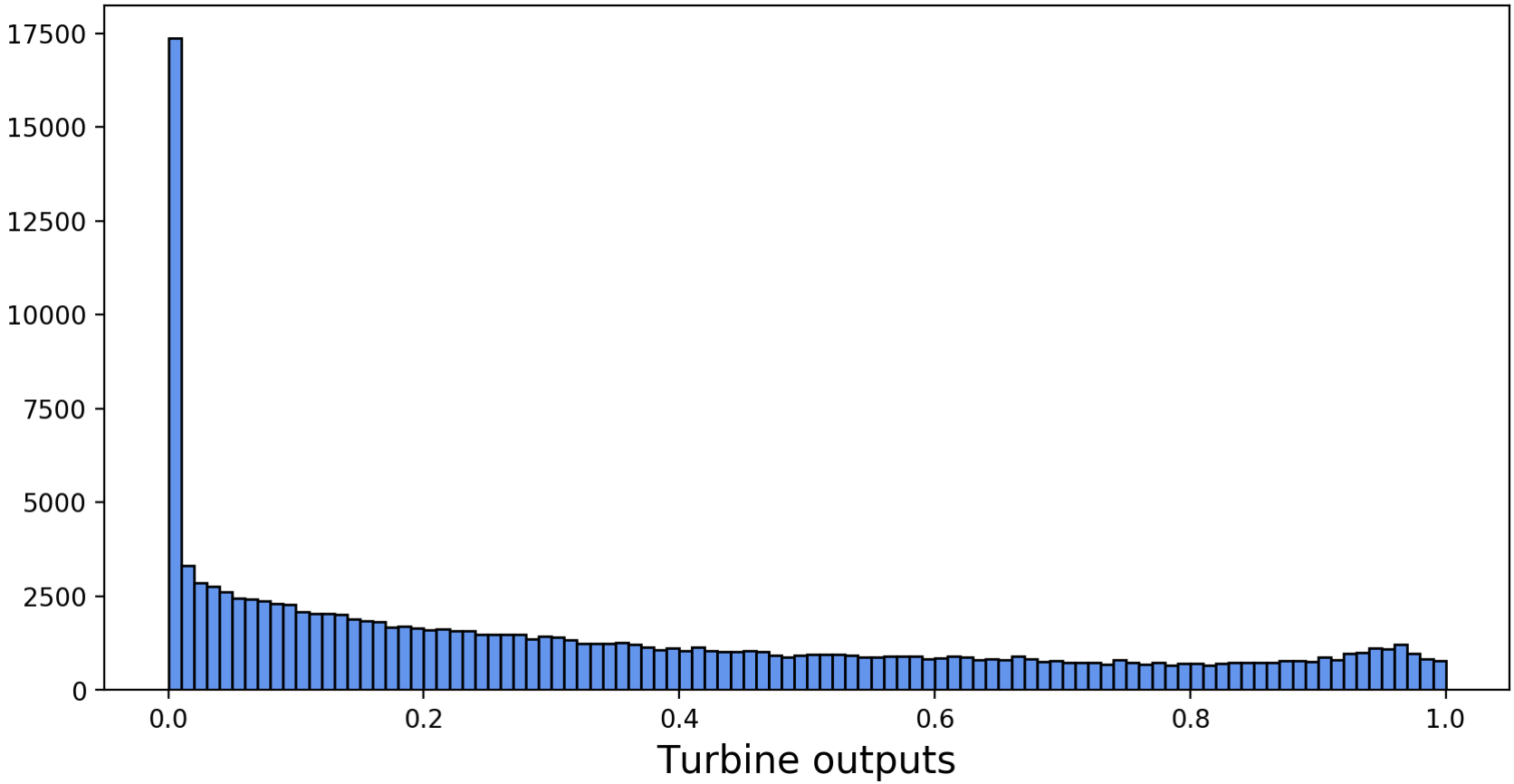
Contributions were evaluated using the root-mean-square error:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\Big[y_{i,\mathrm{pred}} - y_{i,\mathrm{obs}}\Big]^2},$$

i.e. the square root of the average squared difference between predicted and observed turbine outputs.

Exploring the data set

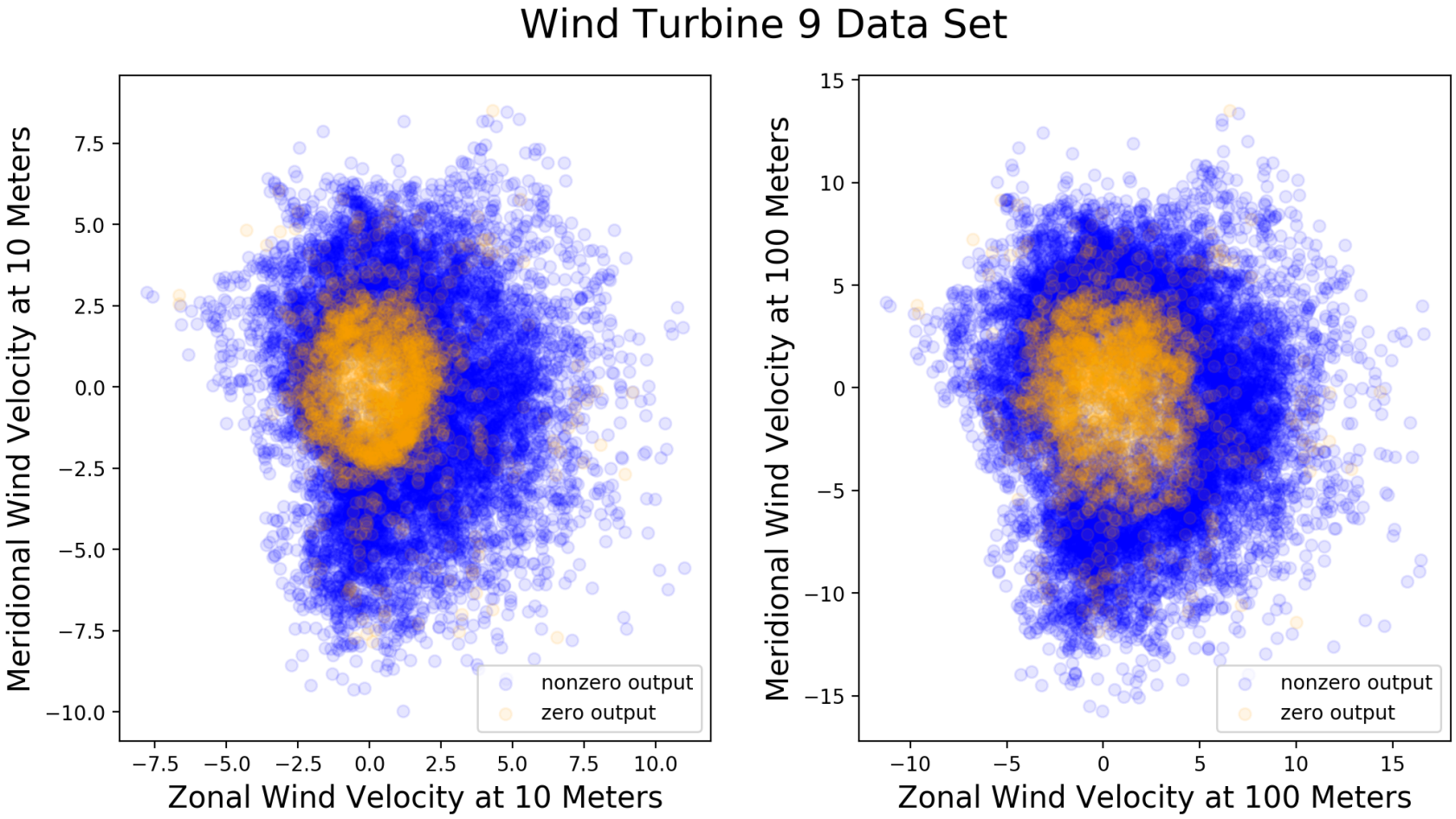
Histogram of turbine power outputs (all turbines together):



- Spike at zero is due to nonzero wind "cut-in speed": this is the minimum wind speed needed for a turbine to start producing power.
- Fraction of turbine outputs that are zero is 8.5% for all turbines combined, but varies from 2.4% for turbine 2 to 17.0% for turbine 9.

Exploring the data set

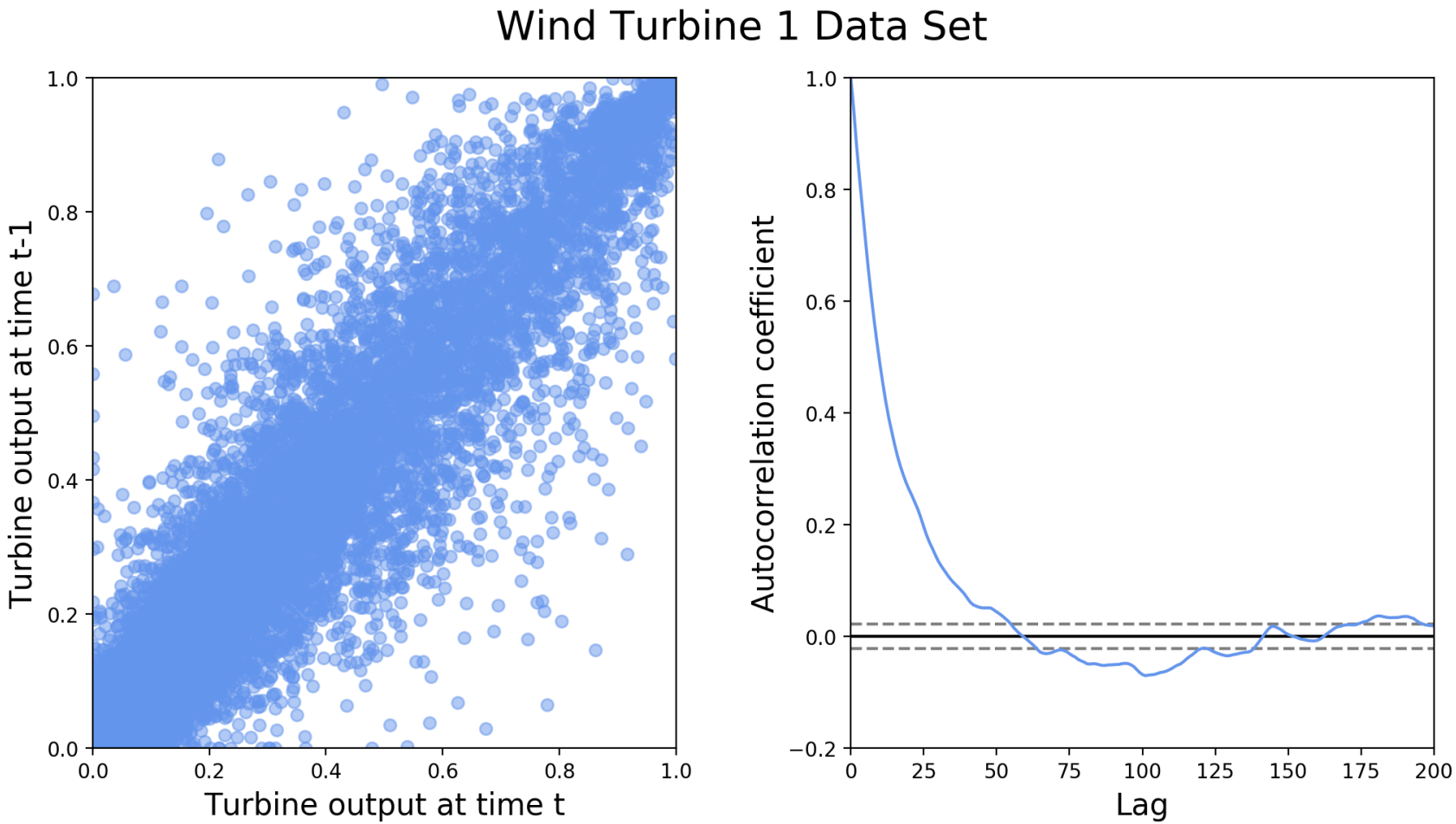
Scatter plots of zonal versus meridional wind velocities:



Note the effect of the cut-in speed!

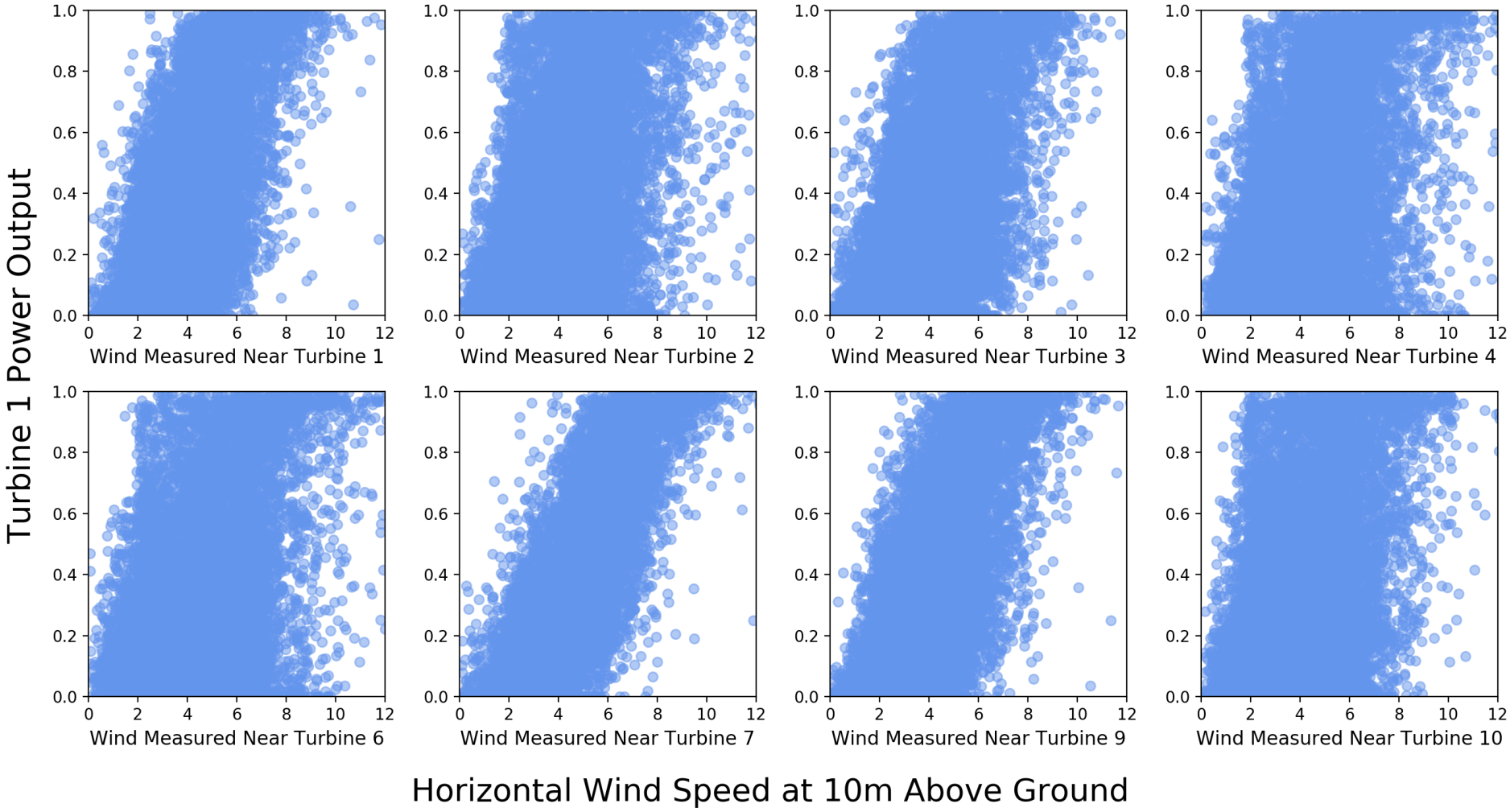
Exploring the data set

Autocorrelation for turbine 1:



Exploring the data set

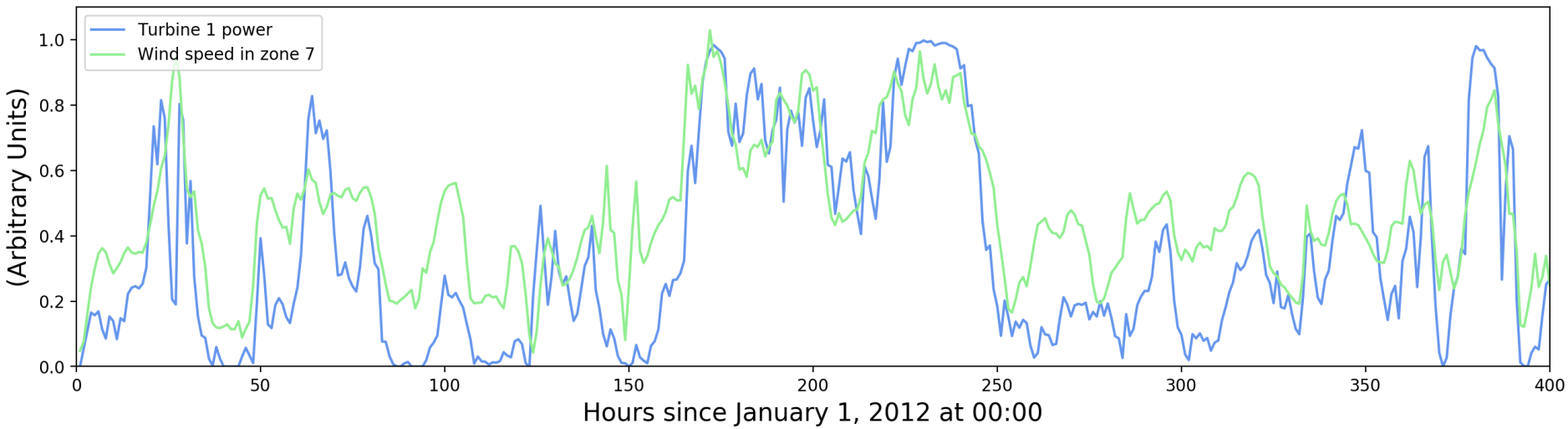
Turbine 1 power versus wind speeds near *other* turbines:



[Note that turbines 4 and 5 share wind measurements, as do 7 and 8.]

Exploring the data set

Turbine-1 power output and zone-7 horizontal wind speed versus time, for the first 400 hours of the data set:



Possible analysis strategies

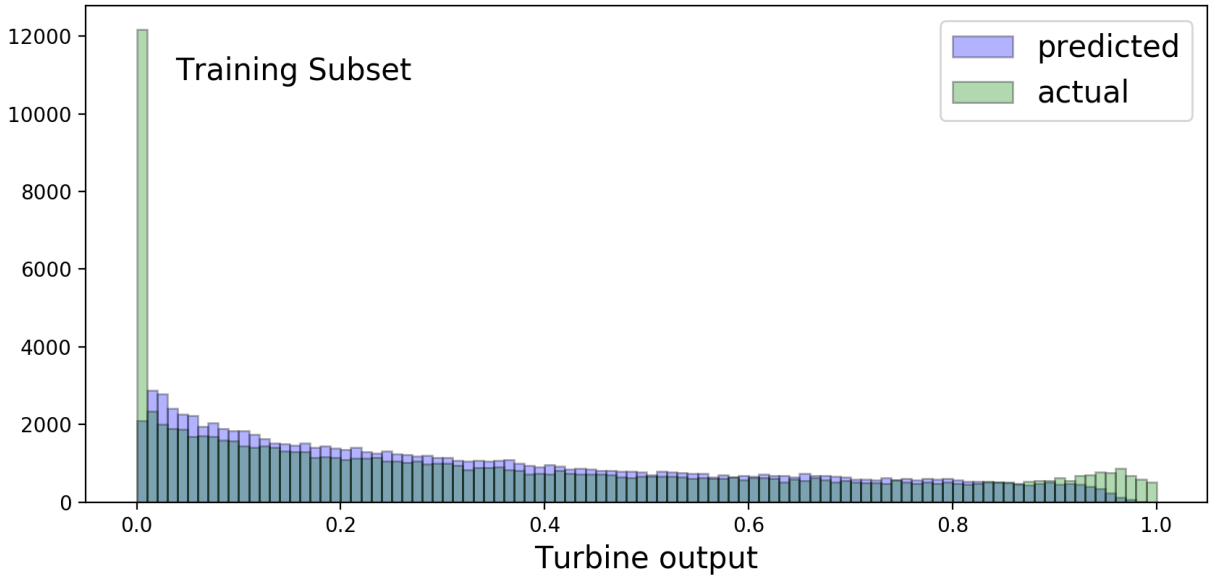
1. Use all 32 wind measurements for predicting each turbine output.
2. Use hour-of-day and day-of-year information from time stamp.
3. Try rolling window average over wind speeds.
4. Try a classifier + regressor combination to handle the spike at zero.

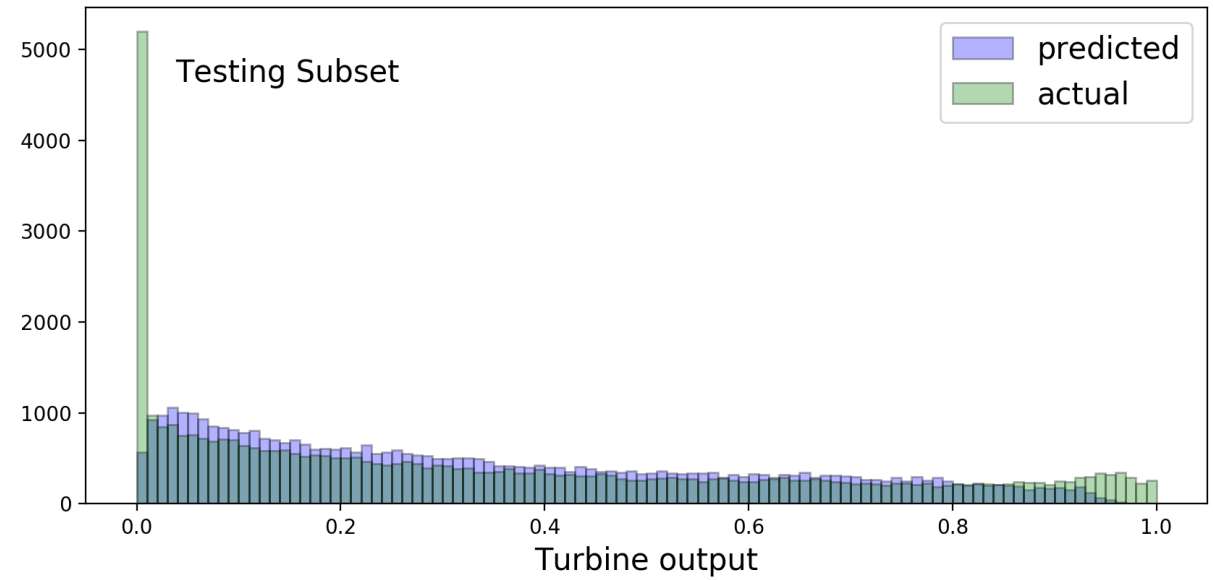
Models I tried:

- random forest (scikit-learn, two models),
- gradient boosted trees (XGBoost, one model),
- generalized linear model (GAMLSS, one model).

Random Forests

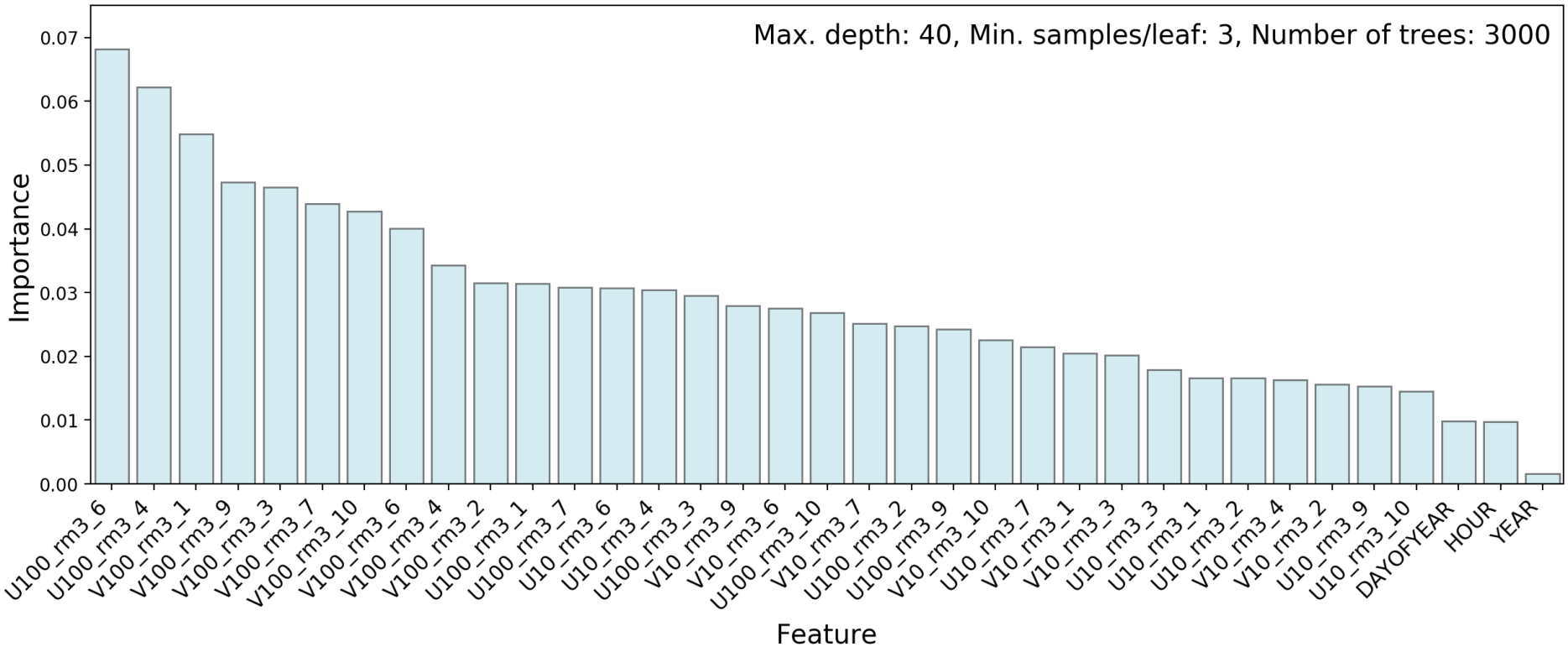
Predicted and observed turbine power output in training and testing subsets:





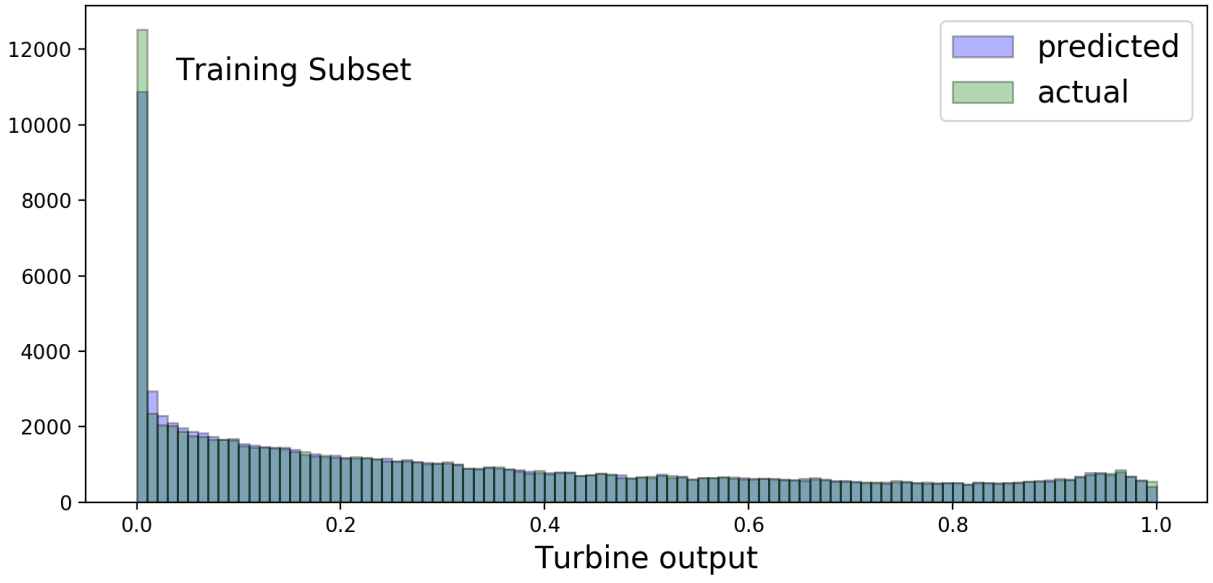
Random Forests

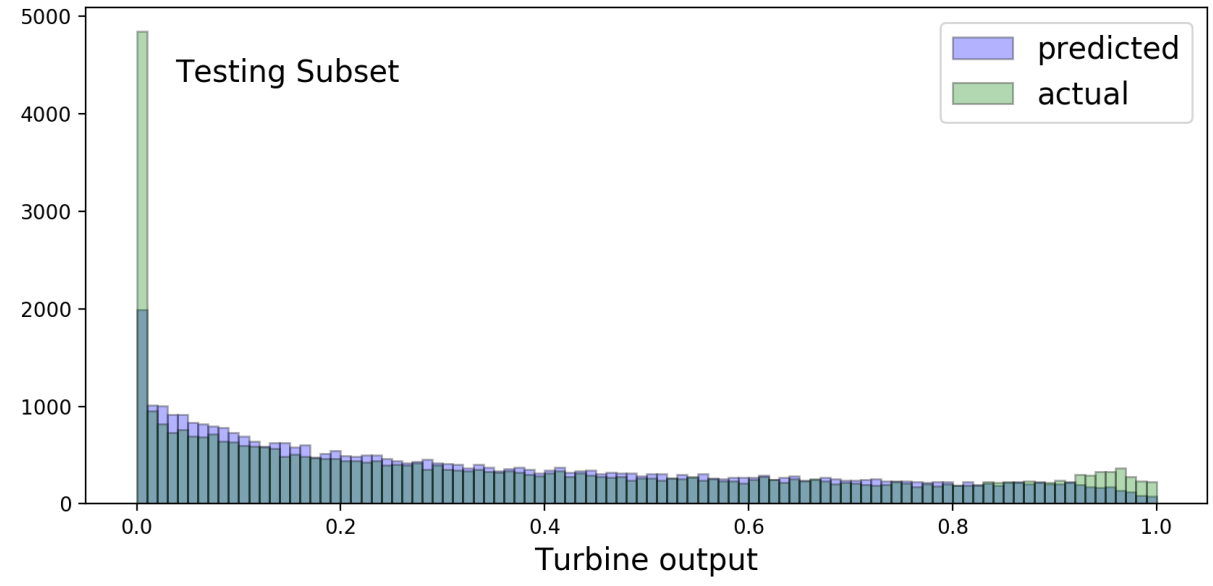
Feature importances:



XGBoost

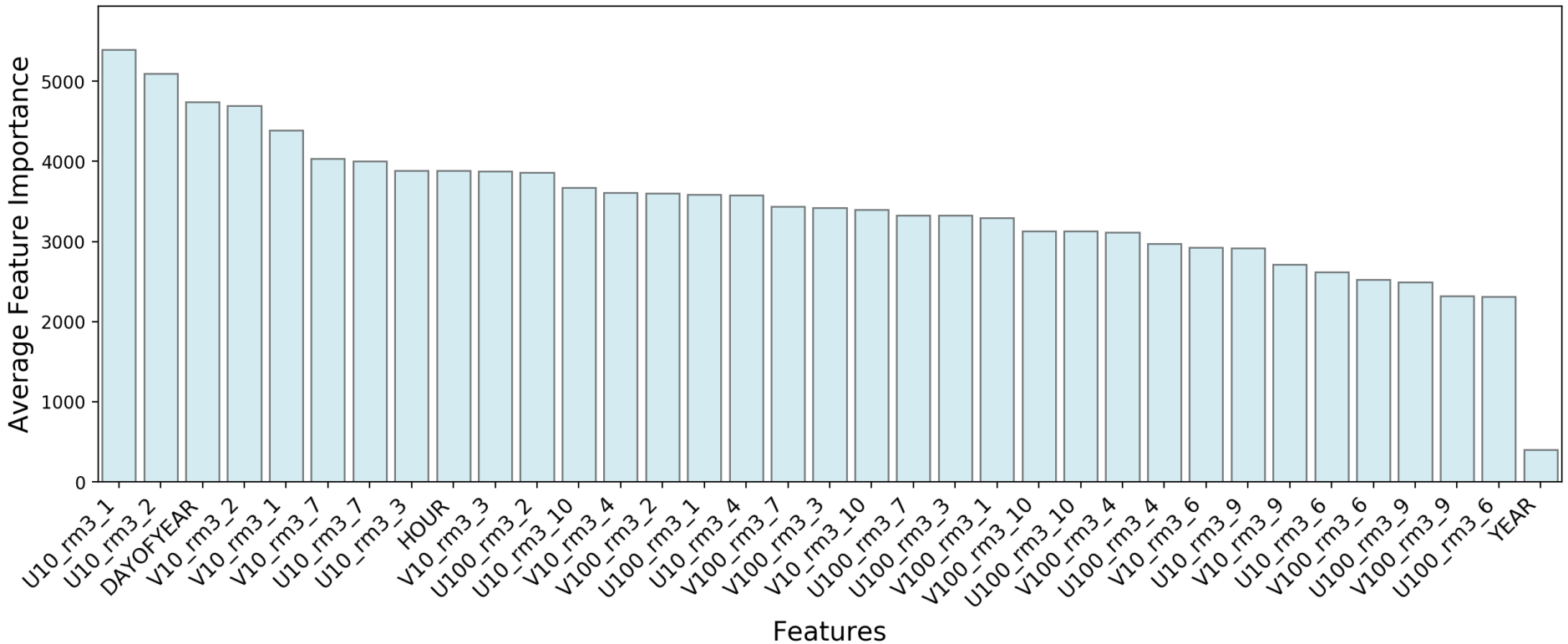
Predicted and observed turbine power output in training and testing subsets:





XGboost

Feature importances:



GAMLSS

Generalized linear model with beta-inflated distribution:

$$f(y \mid p_0, p_1, \alpha, \beta) = \begin{cases} p_0 & \text{if } y = 0, \\ (1 - p_0 - p_1)\, \dfrac{y^{\alpha-1}\,(1-y)^{\beta-1}}{B(\alpha,\beta)} & \text{if } 0 < y < 1, \\ p_1 & \text{if } y = 1, \end{cases}$$

where p_0, p_1 are probabilities and α, β are the beta distribution parameters.
Reparametrize:

$$\mu = \frac{\alpha}{\alpha+\beta}, \quad \sigma = \frac{1}{\sqrt{1+\alpha+\beta}}, \quad \nu = \frac{p_0}{1-p_0-p_1}, \quad \tau = \frac{p_1}{1-p_0-p_1},$$

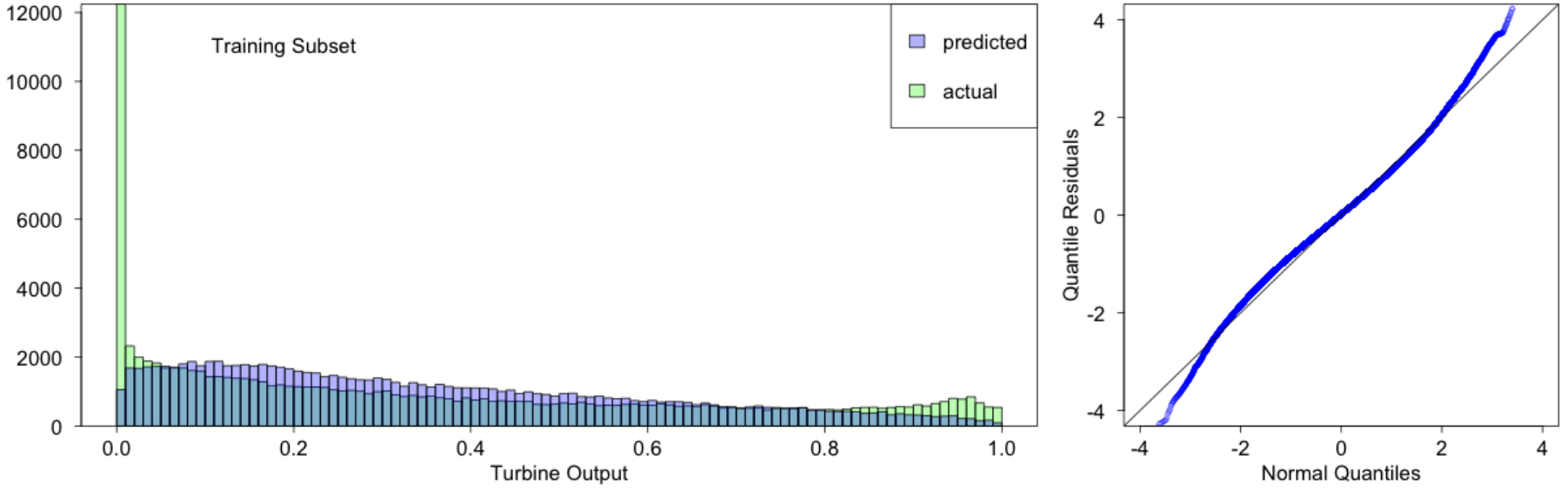
and introduce two links:

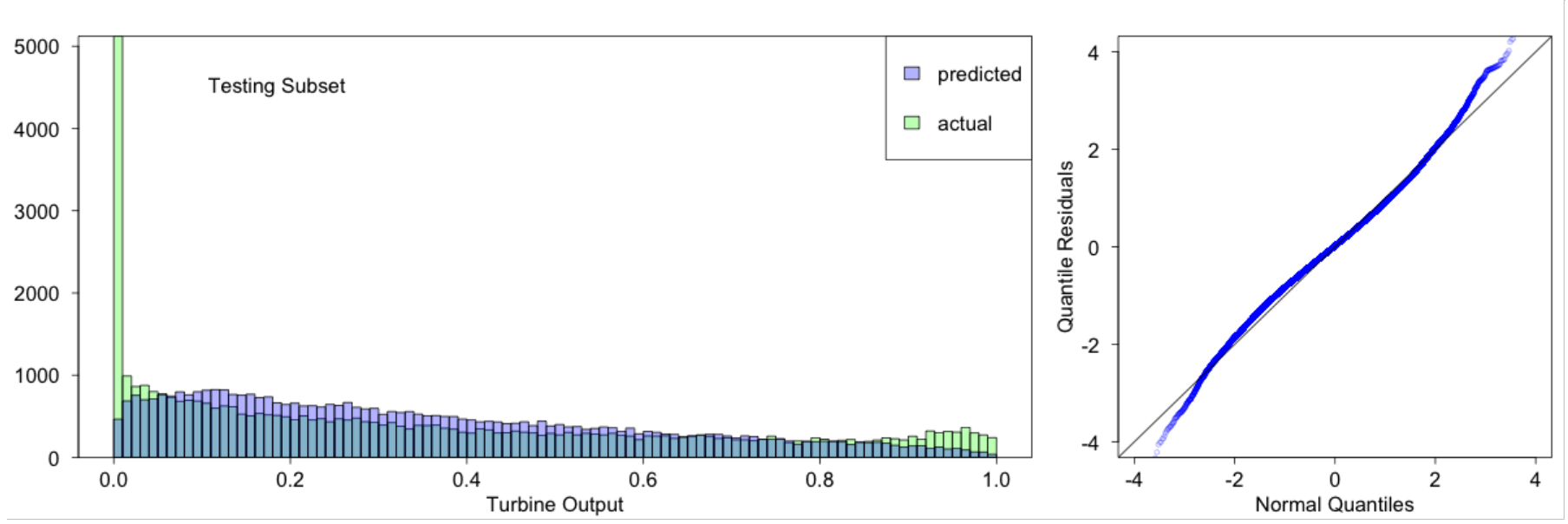
$$\log\left(\frac{\mu}{1-\mu}\right) = c_0 + c_1 X_1 + \cdots + c_p X_p,$$

$$\log(\nu) = d_0 + d_1 X_1 + \cdots + d_p X_p.$$

GAMLSS

Predicted and observed turbine power output in training and testing subsets:





Public leaderboard results

Model	Public Leaderboard RMSE
Random Forest 1	0.180311
Random Forest 2	0.167223
XGBoost	0.167607
GAMLSS	0.167303

Based on this table, which models would you submit for the private leaderboard?

The winner receives an iPad and a job offer...

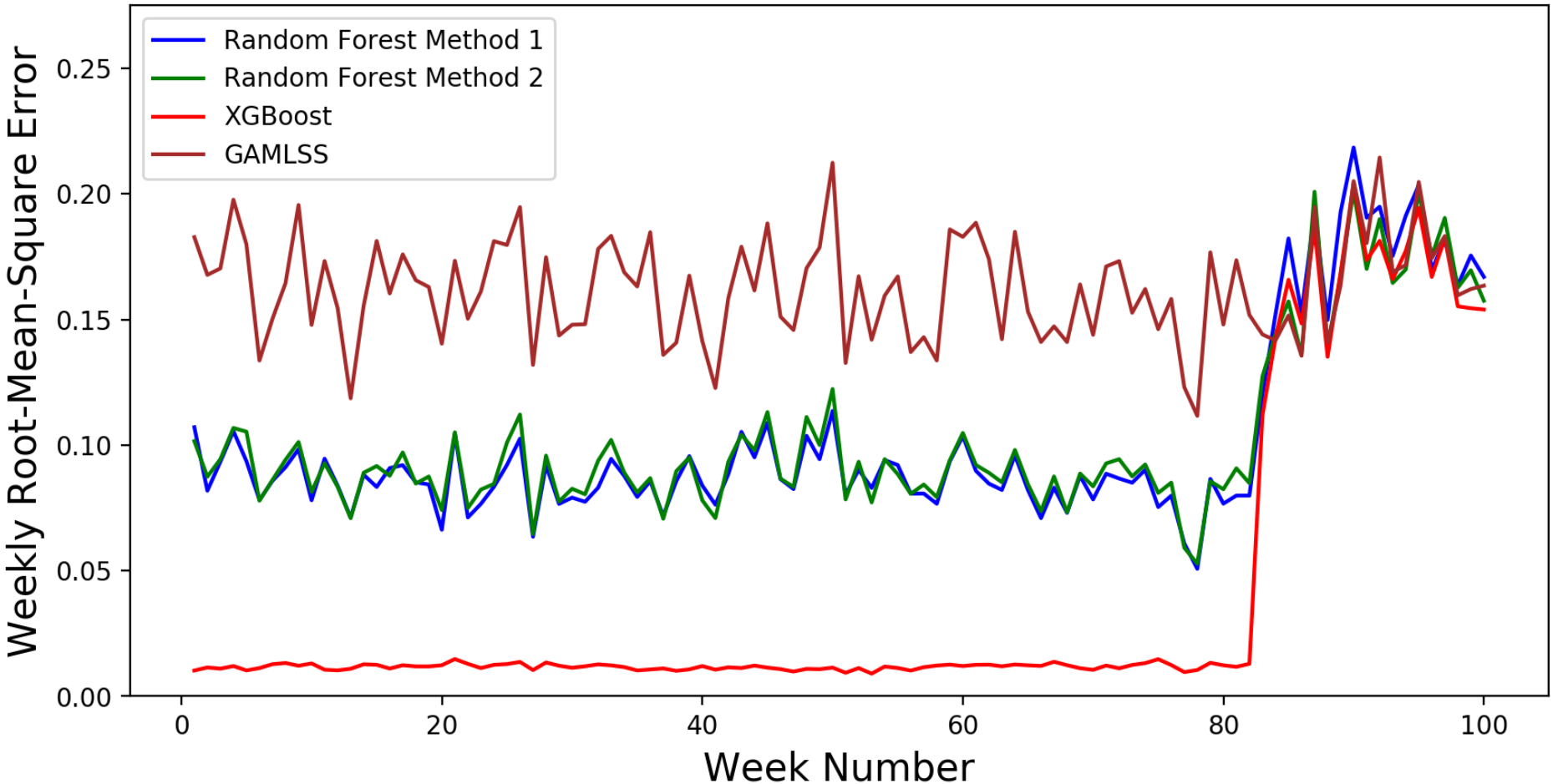
Public and private leaderboard results

Model	Public Leaderboard RMSE	Private Leaderboard RMSE
Random Forest 1	0.180311	0.178977
Random Forest 2	0.167223	0.174105
XGBoost	0.167607	0.168778
GAMLSS	0.167303	0.176077

Compare with winner of hackathon, who obtained RMSE=0.169463 on the private leaderboard.

Public and private leaderboard results

Weekly root-mean-square errors over all 100 weeks of the data set (first 82 weeks are for training data set):



Takeaways

1. XGBoost is remarkable in its ability to model "singular" features such as the peak near zero turbine output.
2. Generalized linear models are very fast to train, but there is clearly a tradeoff to consider between training speed and modeling performance.
3. A machine learning algorithm may perform differently depending on how the data are presented to it. This is the only difference between random forest models 1 and 2.
4. In a hackathon you may want to focus on the performance measure, but keep in mind that the goodness-of-fit of a model may be an indicator of how well the model will generalize.

See http://lucdemortier.github.io/ for more details.