NEM DATA CHALLENGE, 2nd Phase

Sergio Díaz Valdés, September 2017

***Executive Summary**: This report describes the development of a machine learning (ML) model aimed at predicting the power generation output of a Wind Turbine Generator (WTG) using weather variables as predictors. The ML model was trained and tested with 5 months of SCADA records acquired by five WTG's. Previous to training, the raw data was preprocessed to remove anomalous data points. Three ML regression algorithms were tested to built the prediction model. Model performance metrics were obtained and found to achieve high accuracy with respect to six sets of validation data. The ML model was ultimately employed in predicting the power generated by five WTGs for periods of seven days.*

Introduction

Renewable energy generation is one of the fastest growing resources on the electric grid. In 2016, global wind power capacity amounted to about 487 GW [1]. In power systems, balance is maintained by continuously adjusting generation capacity and by controlling demand [2]. If the total output of a wind farm could be predicted with high accuracy, power companies might better plan maintenance of the wind farms and schedule grid maintenance and energy storage systems [3].

This work describes the development of a machine learning (ML) model aimed at predicting mid-term (1-7 days) power generation output of a Wind Turbine Generator (WTG) for a given weather forecast. A Python language script was implemented to read and preprocess 5-months' worth of data collected by the WTG supervisory control and data acquisition (SCADA) system. This historical data was used to train and test the ML model. Additionally, six sets of 24-hr. SCADA records plus 7-day weather forecast were available in csv-format files and used to validate the prediction capabilities of the ML model.

Historical SCADA records

The WTG's SCADA records contained information on speed, velocity, temperature, direction, and pressure acquired by transducers. The SCADA system sampled a total of 28 variables at 1Hz from April to August 2015 on five wind turbines (assets). As preparation for the machine learning model training, the raw data was resampled and averaged at three different frequencies, namely every 1, 2 and 10 minutes. To this effect, the Python script developed during the 1st phase of this Data Challenge was used and its output saved at three distinctive file locations for later use.

Machine Learning Model Development

After exploring the historical SCADA datasets via descriptive statistics and visualizations, a machine learning (ML) regression model was implemented to predict power generation output using wind-speed and ambient-temperature as predictors. Three preliminary models were implemented using a decision-tree (DT), a random-forest (RF), and a gradient-boosting (GB) regression algorithm, respectively. Each model was trained with 70% of the data and tested with the remaining 30%. Metrics used to assess model performance included mean-squared-error (MSE), mean-absolute-error (MAE), and the coefficient of determination (R^2). For brevity, only the latter is shown in this report. Testing the models yielded the preliminary R^2 scores listed in Table 1a, showing that the GB algorithm attains slightly better results than the other two methods; hence GB was used throughout the rest of the work.

Table 1. R^2 (x 100) score per asset and algorithm using: (a) raw and (b) preprocessed historical SCADA data as model input.

(a)

Asset	DT	RF	GB
A001	97.76	97.98	97.99
A002	99.27	99.32	99.34
A003	99.20	99.30	99.30
A004	98.43	98.60	98.58
A005	97.88	98.10	98.14

(b)

Asset	DT	RF	GB
A001	97.77	97.88	97.89
A002	99.26	99.32	99.35
A003	99.16	99.24	99.22
A004	98.49	98.56	98.60
A005	98.06	98.15	98.18

In order to further improve the ML-model scores, a closer visual inspection to the assets characteristic power curve was made. Under-rated-power and near-zero-power data points at wind speeds higher than the cut-in speed were identified and removed from the training data pool. These sub-optimal data points as well as valid power-wind speed data pairs are shown in Figure 1a. The resulting clean datasets were fed into the ML models rendering the new test scores listed in Table 1b. Unexpectedly, this data preprocessing produced limited to null gains towards improving the model performance metrics. Due to time constraints, investigation on the influence of data cleaning on model performance is kept as recommendation for future work.

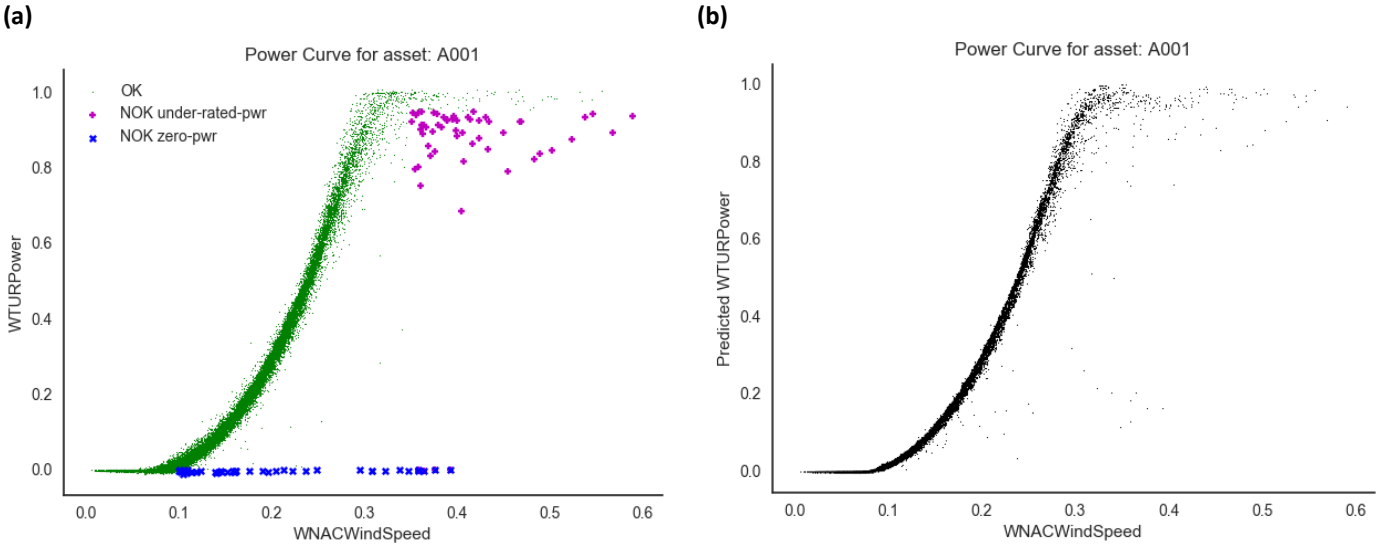


Figure 1. (a) WTG Power curve showing trend data pairs (in green) together with under-rated and near-zero power points above the cut-in wind speed. (b) Predicted Power curve for asset A001

Figure 1 (b) displays the predicted Power curve for asset A001, where a much lower dispersion of the data pairs can be observed. Correspondingly, Figure 2 depicts a scatter plot of the predicted power vs. the actual power produced by asset A001, where an overall good agreement is perceived. For intelligibility, Figure 2 shows only few days in August out of the five months available.

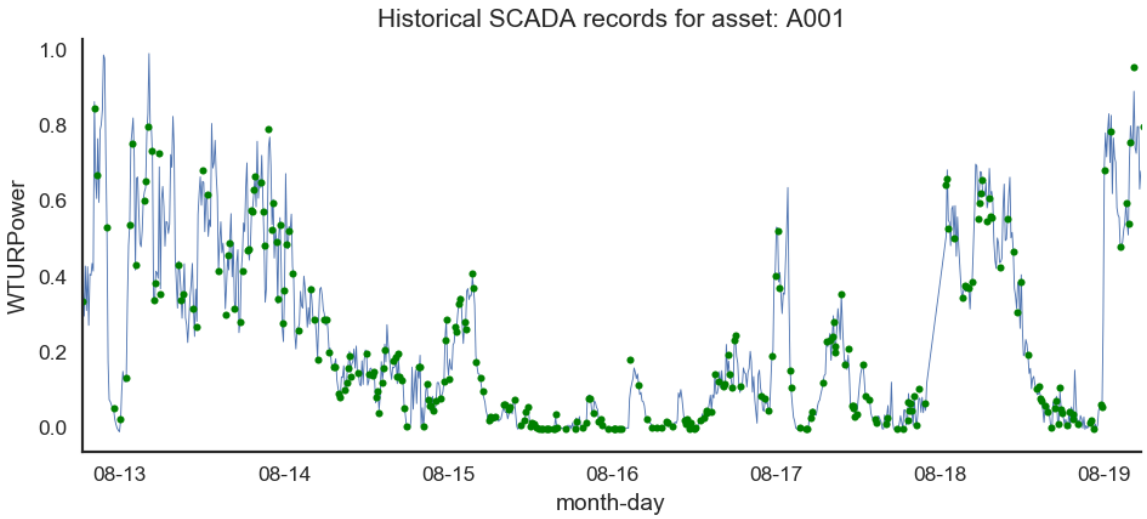


Figure 2. Predicted (dots) vs. power output signal (continuous line) for asset A001 during August 2015.

Machine Learning Model Validation

The machine learning predictive model was validated using the 24-hr. SCADA data available for each asset. Tables 2a-2c list the 100 x R^2 score obtained by comparing real vs. predicted power values per asset and evaluation date. Values shown in Table 2, from left to right, were obtained using the same ML algorithm but trained with historical SCADA data resampled from the original 1Hz frequency to 10, 2, and 1 minute intervals, respectively.

Models trained with data resampled at 1 and 2 minute frequencies exhibited a slight improvement over the 10-minutes case, with a global average 100 x R^2 value of 98.83 for Table 2a, 98.95 for Table 2b, and 98.91 for Table 2c. The calculation of a global mean value per table excluded the results corresponding to evaluation date 2015-09-09 (gray-shaded columns), since on that day the SCADA system failed recording data in a continuous manner, thus requiring extra care when computing deviations between predicted vs. real power values.

Table 2. R^2 (x 100) score per asset and validation date for each ML algorithm trained using historical SCADA records resampled at frequencies: (a) 10 minutes, (b) 2 minutes, and (c) 1 minute.

(a)

Asset	sep-01	sep-09	sep-17	sep-25	oct-03	oct-10
A001	99.39	77.77	99.77	99.05	98.63	99.73
A002	99.17	-101.68	99.40	98.10	98.45	99.71
A003	99.30	83.07	99.83	98.42	96.88	99.54
A004	99.61	94.59	99.59	97.88	98.06	98.58
A005	99.17	90.37	99.54	97.00	96.49	99.52

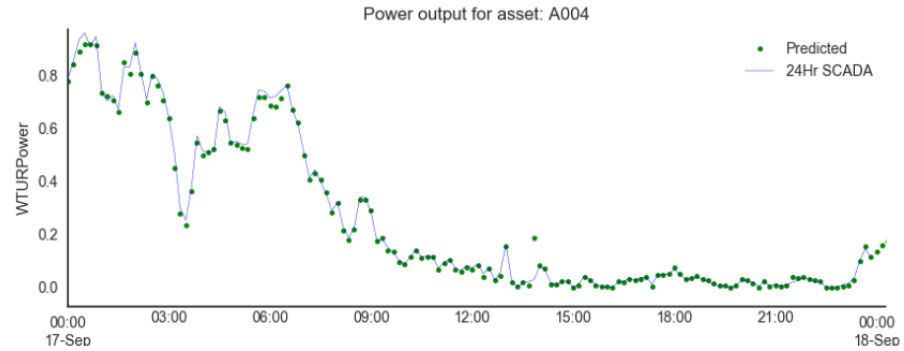
(b)

Asset	sep-01	sep-09	sep-17	sep-25	oct-03	oct-10
A001	99.30	77.32	99.77	99.15	98.95	99.80
A002	99.20	-91.62	99.41	98.54	98.57	99.76
A003	99.14	83.89	99.82	98.74	97.32	99.55
A004	99.52	94.74	99.62	98.22	98.32	98.74
A005	98.75	90.82	99.78	97.27	96.96	99.66

(c)

Asset	sep-01	sep-09	sep-17	sep-25	oct-03	oct-10
A001	99.19	78.15	99.77	99.24	98.91	99.77
A002	99.18	-91.79	99.41	98.46	98.52	99.65
A003	98.98	83.70	99.79	98.76	97.14	99.56
A004	99.39	94.89	99.59	98.22	98.27	98.83
A005	98.88	91.27	99.66	97.45	96.57	99.59

(a)



(b)

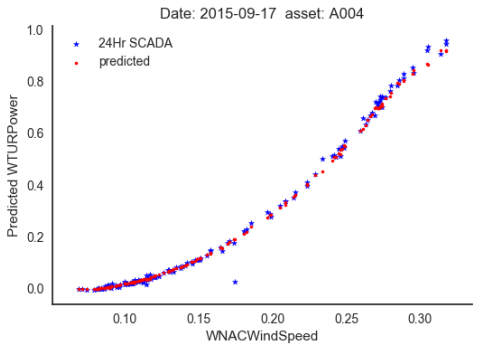
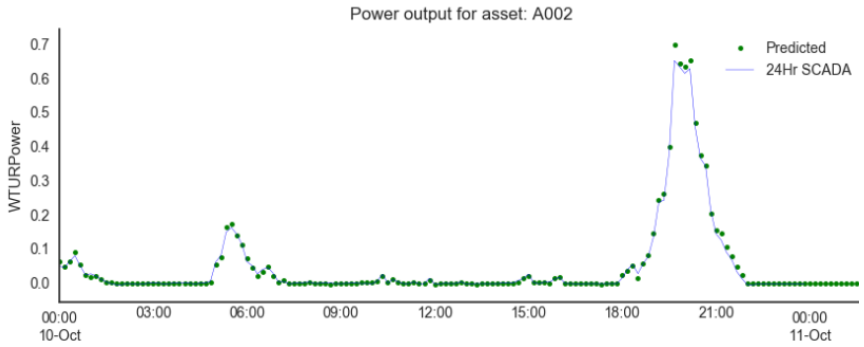


Figure 3. (a) Scatter plot and (b) Power curve showing both, 24-hr. SCADA and predicted power on evaluation date 2015-09-17.

(a)



(b)

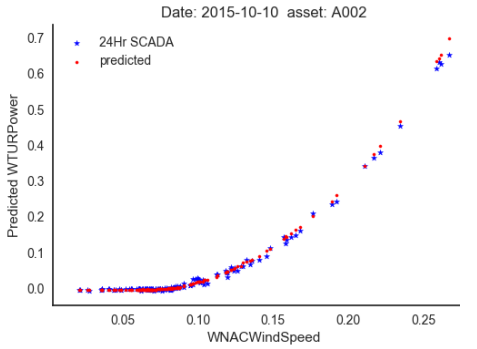


Figure 4. (a) Scatter plot and (b) Power curve showing both, 24-hr. SCADA and predicted power on evaluation date 2015-10-10.

Figures 3 and 4 display a very good match between real vs predicted power values. In accordance to its corresponding time-power scatter plot, the power curve shown in Figure 3b has a higher concentration of data points at the upper-right end of the curve. Contrastingly, Figure 4b has a higher concentration of points at the lower-left end of the curve, most likely due to the prevalence of low wind speeds throughout that date.

Evaluation Weather Forecast

After the prediction model was built, efforts were focused on preparing a 10-minute frequency, 7-day weather forecast for each one of the six evaluation dates defined in the Data Challenge. The main input to generate such projections came in the form of csv-format files containing hourly weather information for a number of 7-day periods between September and October 2015. In summary, the Python code implemented to produce the weather forecasts executed the following tasks:

1. Create an empty table (data-frame) containing a row every hour for 7 days, with columns defined for wind-speed, ambient-temperature, range, forecast timestamp, range equivalent date-time, and asset.
2. Copy the 7-day forecast provided as csv-format files onto the empty table created in step 1.
3. Calculate mean values across assets for both variables: wind speed and ambient temperature.
4. Use then calculated mean values to fill-in missing data points found while exploring the csv-files content.
5. Create a second empty table with a row every 10 minutes for 8 days (24-hr. SCADA + 7-days).
6. Populate the empty table from item 5 with the data generated in steps 1 to 4.
7. Populate the 10 minutes table with weather variables from the 24-hr. SCADA data.
8. Interpolate weather variables of the resulting table from hourly values to estimates every 10 minutes.

Step 3 of the task list was considered necessary to account for some missing values found in the evaluation 7-day forecasts. Such information gaps are shown graphically in Figure 5 where discontinuities in the colored lines represent missing data at given date/time combinations.

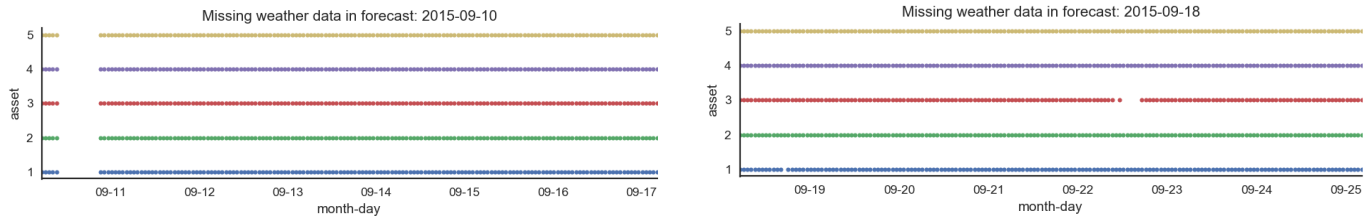


Figure 5. Missing values in the weather forecast starting on 2015-09-10 (left) and 2015-09-18 (right).

An illustration of the final output produced by the programming script is shown in Figure 6 and 7, where the wind-speed and ambient temperature forecast, respectively, are presented for two of the six evaluation dates.

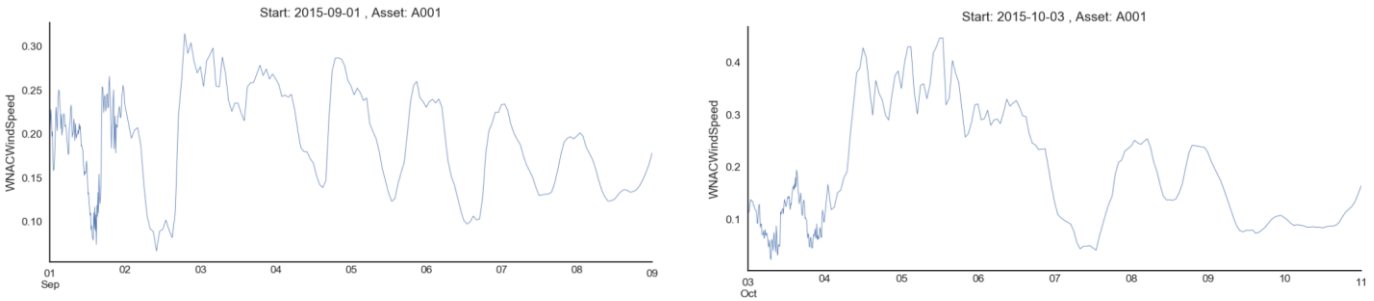


Figure 6. 24-hr. SCADA plus 7-day wind-speed forecast for periods starting on (left) 2015-09-01 and (right) 2015-10-03.

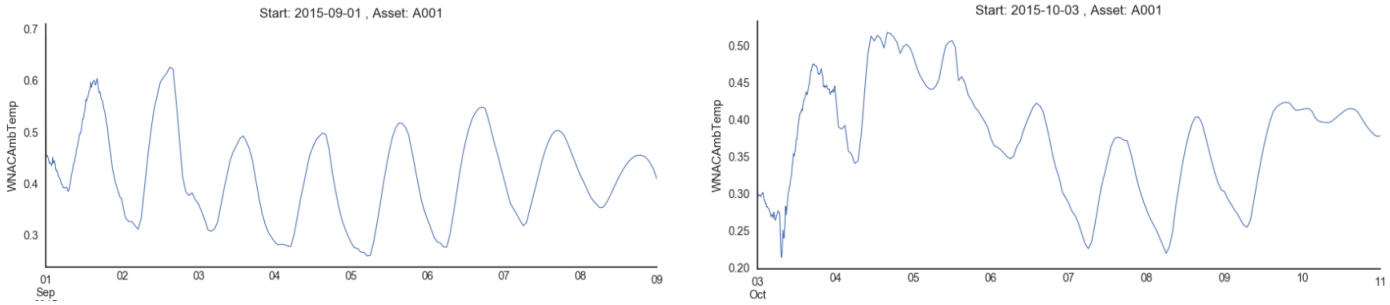


Figure 7. 24-hr. SCADA plus 7-day ambient temperature forecast for periods starting on (left) 2015-09-01 and (right) 2015-10-03.

Power Generation Prediction

Power generation estimates were easily obtained by feeding the trained ML model with time-series of ambient temperature and wind speeds. Figure 8 shows predictions for power and energy production corresponding to three of the six 7-day evaluation forecasts. Values shown in figure 8 represent the arithmetic mean across the five assets.

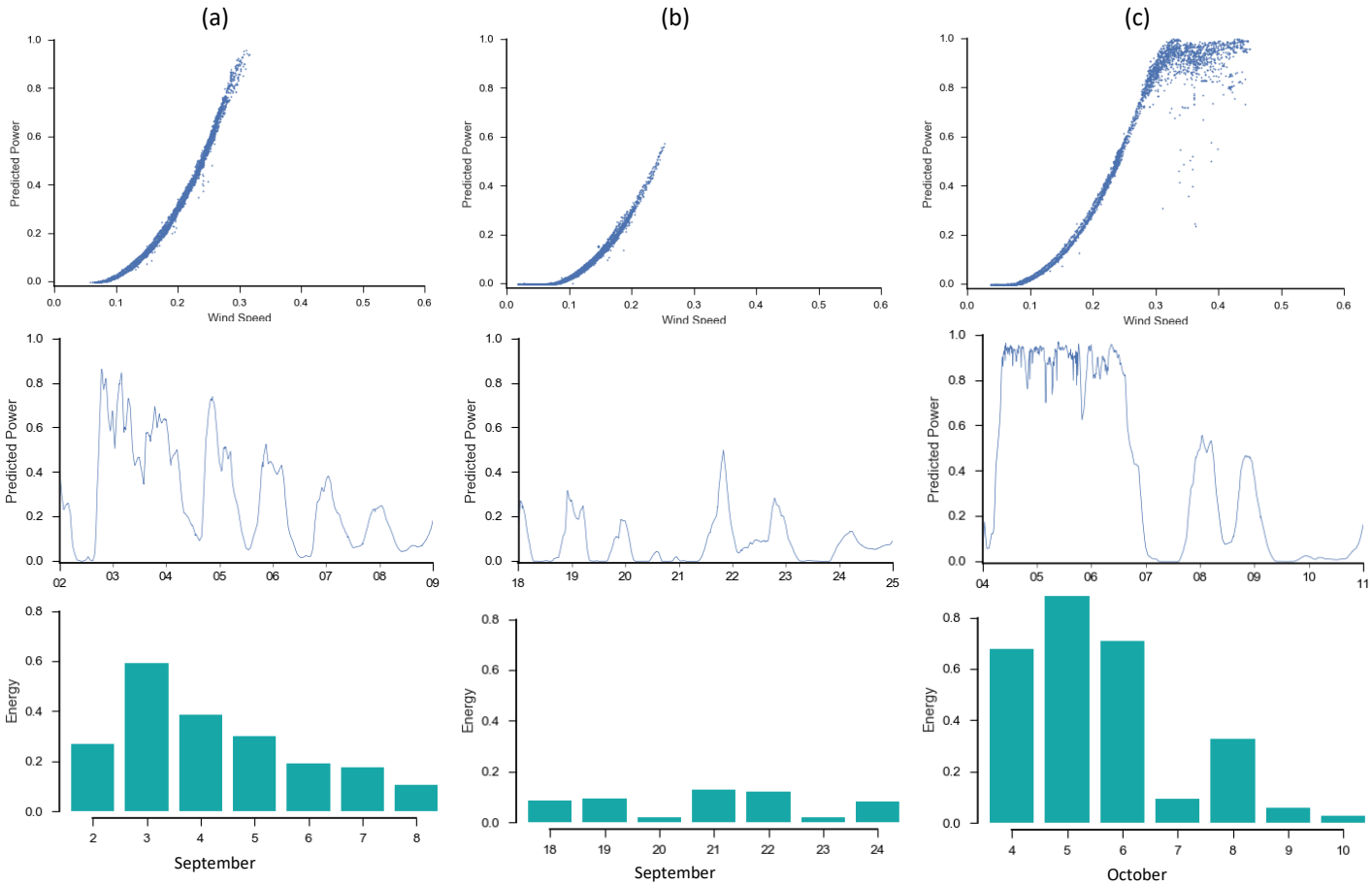


Figure 8. 7-Day prediction showing, from top to bottom: Power Curve, Power output (assets mean), and daily Energy Produced (assets mean). Plots correspond to week commencing on: (a) Sept-02, (b) Sept-18, and (c) Oct-07-2015.

Concluding remarks

A machine learning model has been implemented capable of predicting the power generation output of a Wind Turbine Generator (WTG) using wind speed and ambient temperature as predictors. A Gradient-Boosting regression algorithm was selected as the main gear of the predicting module. Model performance was quantified using common error metrics and found to be in good agreement with respect to real measured data.

In spite of the apparent good correlation found between predicted and measured data, some uncertainties remain with respect to the truthfulness of the weather forecast as proposed in this work. Future work may be focused on establishing a correlation between the historical weather variables recorded by the SCADA systems and their associated weather forecasts. Such insight can then be applied to derive correction factors to adjust mid to long term forecasts.

References

[1] GWEC Global Wind Statistics 2016 (PDF). GWEC. 10 February 2017.
[2] A.M. Foley, P. G. Leahy, A. Marvuglia, E. J. McKeogh. Renewal Energy 37 (2012) 1-8.
[3] Z. Liu, W. Gao, Y.H. Wan and E. Muljadi. Conference Paper NREL/CP-5500-55871 August 2012.