The following material was released on December 8, 2021 by the Windscape.ai team. I'm excited to share it with you as it summarizes our pilot results demonstrating the ability to produce wind alerts from the low-cost pressure sensor network. This is a big step in validation of our value proposition of a low-cost method for increasing wind farm output. More results are coming soon from this pilot data.

Our next step is a couple of pilots at commercial wind farm owners' sites where we'll deploy next-gen AI analysis and sensors.

Windscape.ai Pilot 1 Results Summary

Abstract

The Windscape.ai Pilot 1 was designed to explore the basic project hypothesis that 10-60 second wind speed and direction forecasts at a particular location can be obtained by a machine learning (ML) model and current-time air pressure readings from a distributed array of low-cost sensors. These short-term wind forecasts could be used by wind farm operators to increase the efficiency of their turbines and reduce gust-induced loading and failures. As a proof of concept, we set up a reference anemometer and a network of 20 pressure sensors in a test site near UC Davis. This site serves as a small scale representation of a wind farm, where we would utilize existing reference anemometers on turbine nacelles and met towers, and install our own pressure sensor array around the wind farm. We gathered seven weeks of data, performed some exploratory analysis, and developed a simple ML model to predict future anemometer wind speed readings as a function of earlier pressure readings from our sensor array. The pilot gathered data from 20 sensors including an anemometer on a tower for seven weeks. Our ML analysis has shown the ability to predict wind speeds from the sensor data.

December 8, 2021

Windscape.ai

Contents

Overview

We are taking advantage of advances in edge AI and newly available low-cost pressure sensor technology to explore a new approach to providing preview data for wind farms. Currently, almost all wind farm turbines wait until after wind change events arrive to adjust blade pitch and nacelle direction (yaw), and consequently turbine settings always lag optimal settings. This results in lost energy and accumulated damage to gearboxes, blades and towers. We are targeting the value proposition for wind farms of increased energy production and lower O&M costs by enabling control systems to preemptively adjust settings ahead of wind changes.

We do this with a methodology employing AI pattern analysis performed on data coming from an array of very-low-cost pressure sensors mounted at fence post height in a wide area outside the wind farm. AI pattern recognition is performed on this system of pressure and anemometer data to recognize patterns that indicate movement of air across the landscape. These changes relate to wind changes that will occur at the turbines 10 to 60 seconds later. Commercial systems at wind farms would send real time wind alerts to wind farm operators and control systems.

In this pilot, reference data for the AI pattern correlation comes from an anemometer on a 17m tower representing a wind turbine. In commercial application at a wind farm, we would use existing anemometers on the wind turbines and existing wind farm meteorological towers from 50m to 80m tall. This project used 20 pressure sensors, whereas a commercial installation could have 50 or more.

This approach is 5x-10x less expensive than other technology for creating wind farm preview data, such as doppler LIDAR, and promises to develop a picture of coming wind speed and direction changes from a wider area. We believe this approach has many other applications where warnings of wind changes are important, including tall crane operations, airports, eVTOL sites, rocket launch sites and recreational flying.

This process is broadly patented under [US8413500B2.](US8413500B2.)

Pilot Strategy

For this seven-week Pilot, we collected data and developed and tested the preliminary AI analysis. We used the actual data to run various AI analysis scenarios for test and demonstration.

The Pilot had the following main accomplishments:

- Developed cooperation with the UC Davis Atmospheric Science Research group to utilize equipment and facilities at their Campbell Tract meteorology site. [map and photo]
- Set up a test with real-time collection of data from one anemometer and sensors distributed on a wide area.
- Maintained the system for 7 weeks.
- Collected data on site and remotely.
- Conducted and wrote up analysis statistically and using AI methods.

Design

Test Apparatus Design

The test apparatus was deployed at the Campbell Tract run by UC Davis Atmospheric Research Group. The test apparatus was based on open source protocols and readily available, off the shelf components. Nothing new was invented to configure the hardware, which kept design and maintenance low cost. Some proprietary innovation went into the wireless mesh network protocols to efficiently handle the 1 Hz data from 20 sensors over a low power wireless network. There was also considerable inventiveness in the sensor node box configuration to keep costs down.

The Davis site was attractive for a number of reasons. It had an anemometer and meteorology communications infrastructure in place. The UC Davis Atmospheric Research Group was interested in the project and particularly helpful with set up, communications and technical assistance. The Davis area has an acceptable wind distribution that time of year consisting of relatively mono-directional winds from the south and wind speeds high enough to be helpful for analysis. See Figure 1 below. This wind regime allowed us to adequately represent a wind farm without being too complex.

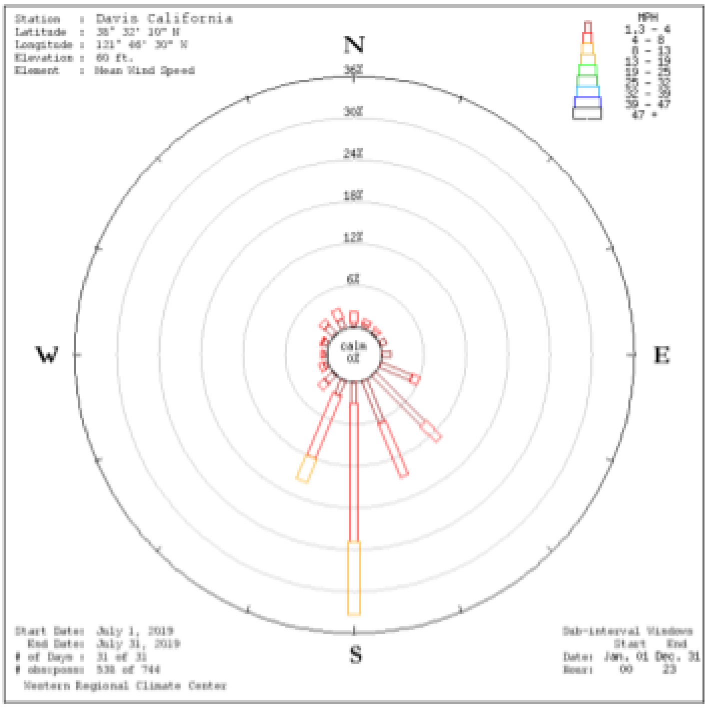

Figure 1: Davis area wind rose

20 sensor nodes were deployed on fence posts around the Campbell Tract, and 19 continued to work throughout the Pilot. Each sensor node includes sensors for pressure, temperature and battery life included in a custom built Arduino board. Wifi mesh networking capability was programmed into the Arduino board which allowed the distributed sensors to automatically set up a wireless mesh to communicate to the Gateway central data collection point (described below).

Pressure readings were taken by the Bosch BMP280 absolute barometric pressure sensor. Figure 2 below. The BMP280 has relative accuracy of ±0.12 hPa and absolute accuracy of ± 1 hPa. Windscape.ai's analysis depends on relative pressure measurements. The BMP280 has an operating range for pressure of 300 - 1100 hPa and temperature: -40 to 85°C. These are very low cost components running about $2 each on a retail basis targeted at mobile phones, watches and other high volume markets. Bosch continues to enhance the product line with the BMP388 model offering enhanced accuracy now entering the market.

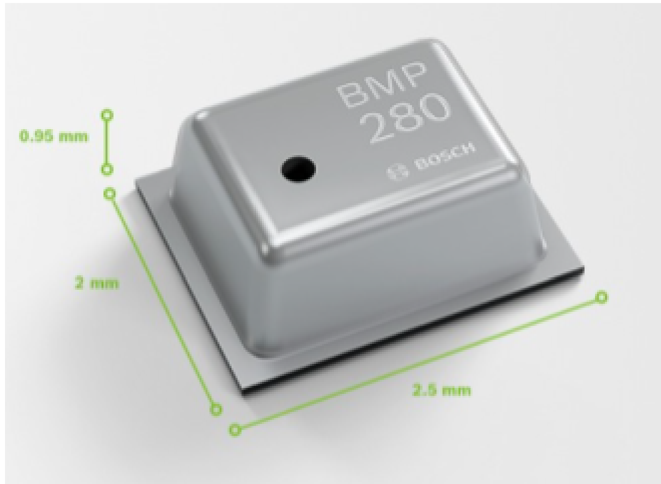

Figure 2: BMP 280 Pressure Sensor

Temperature and battery readings are only utilized to monitor the health of the sensor node box. Temperature is not meant to represent weather conditions.

These “Gen 1” sensor nodes were battery powered for simplicity. The batteries would last about 1 week before needing a recharge. Gen 2 sensor nodes will be solar powered.

Physically, each sensor node consists of a low-cost IP65 rated equipment box modified with an open bottom to transmit pressure changes and an antenna for the wireless mesh network. An example is shown in Figure 3. Figure 4 shows an example of how Gen 1 sensor boxes were mounted on fence posts.

Figure 3: Sensor node configuration

Figure 4: Typical sensor node installation at the Campbell Tract

Anemometer

A cup anemometer and wind vane were mounted on a 17m pole attached to the Campbell Tact control building and utilizing power from there. The anemometer provided continuous data wirelessly through the system gateway.

Gateway

The system gateway was configured with an antenna and a Raspberry Pi server to manage wifi communications with the nodes mesh network and to log data. It utilized power over ethernet (POE) and direct ethernet data communication with the system hub. This gateway was custom built for this pilot, and it can serve as a model for a commercial product.

System Hub

A small Raspberry Pi server box was configured to reside inside the control building - Figure 5. It consisted of a 4”x4” screen - Figure 6 - networking, mobile data connection and a UPS. It also served as a secondary data repository. The System Hub could be accessed onsite via keyboard and the screen or accessed remotely. Similarly, data can be downloaded manually onsite via USB drive or accessed remotely.

Figure 5: System Hub layout

SensorNET Field test Time - - 21.33.25 Gateway - - sensornet.gotdns.com Proxy - - 192.168.1.1

Anemometer	ID - Mesh	health	Tx #	Wind	Dir	Diff	Tx fails	Cycle (ms)	vBat
1	006 - 0002	12345678901234	300412	6.1678	247.5000	0.0000	1171	799	4.867
Sensors	ID - Mesh	health	Tx #	Temp F	Pres mBar	Diff	Tx fails	Cycle (ms)	vBat
2	020 - 0003	12345678901234	34665	71.5340	1008.0699	0.0100	715	799	5.028
3	028 - 0031	12345678901234	31668	69.3200	1008.1479	0.0159	567	799	4.28
4	032 - 0033	12345678901234	33575	70.7600	1006.9651	0.0123	952	799	4.241
5	036 - 0000		29	87.2480	1007.5494	-0.0104	7	799	4.229
6	044 - 0034	12345678901234	27897	70.2380	1007.6016	-0.0016	1937	799	4.222
7	048 - 0005	12345678901234	32747	70.0940	1006.6945	-0.0052	1180	799	4.171
8	052 - 0004	12345678901234	43695	66.2960	1006.9788	0.0044	545	799	4.177
9	056 - 0323	1234567	28785	65.9180	1007.0432	-0.0146	1650	799	4.164
10	060 - 0001	12345678901234	34894	64.7840	1007.4473	-0.0195	273	799	4.171
11	064 - 0043	12345678901234	186054	65.8640	1007.8566	0.0149	1143	799	4.203
12	076 - 0433	12345678	33505	29.0000	0.0000	233.1023	950	799	4.229
13	080 - 0023	12345678901234	188185	68.0780	1006.9374	0.0050	807	799	4.106
14	096 - 0044	123456789012	32655	29.0000	0.0000	0.0000	450	799	4.158
15	104 - 4323	12345678901234	56997	65.5220	1008.5848	-0.0120	7436	799	4.19
16	112 - 0024	12345678901234	35648	66.9440	1006.5078	-0.0012	299	799	4.19
17	128 - 0021	12345678901234	34902	68.2580	1008.1330	-0.0007	428	799	4.183
18	132 - 0042	12345678901234	193306	68.6180	1006.4286	0.0000	2520	799	4.222

18 sensors	1329619	1103.6438	15358.4459	23030	version 6-13-2021
	error rate	avg temp	avg pres	---------	---------
21.33.25	1.7321	61.31	853.25	copyright	SensorNET2020

Figure 6: Example of data summary display

Field Test Design

This pilot was set up in cooperation with the UC Davis Atmospheric Science Research group at the Campbell Tract meteorology site - Figure 7. The site doubles as an agricultural test tract and therefore includes large open areas used for low-height crops and fallow fields and fences dividing them. There is a small control building at the center of the property that houses meteorological data collection servers - Figure 8.

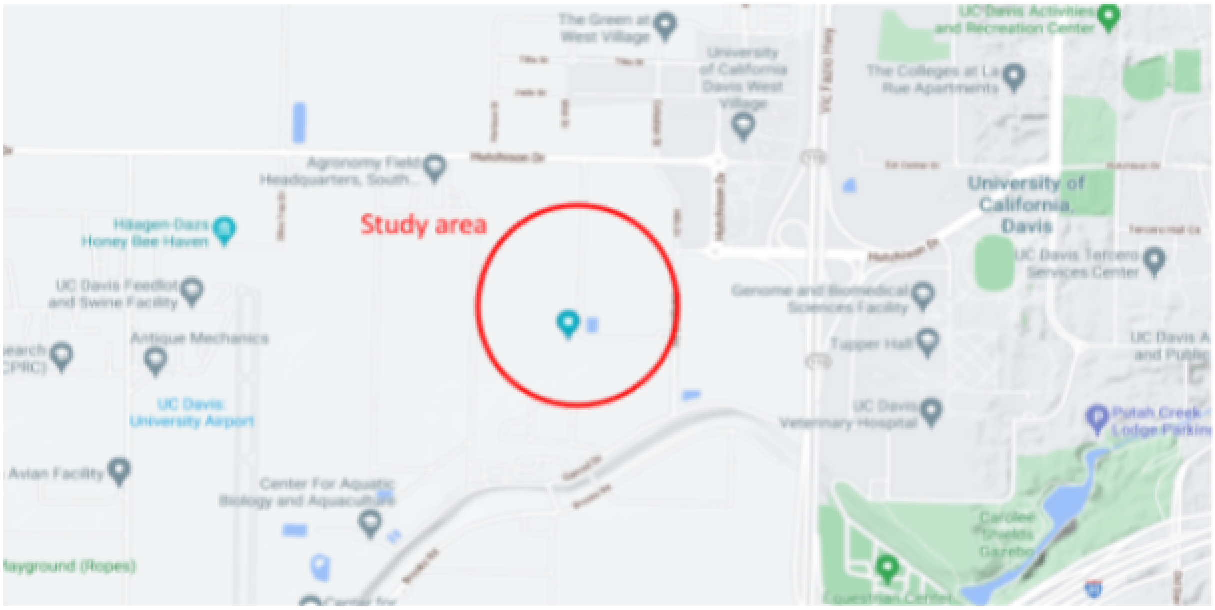

Figure 7: Campbell Tract study area near UC Davis

Figure 8: Site and control building

The test setup consisted of 19 pressure sensors at fence post height distributed throughout the surrounding area, up to 300m from the reference anemometer - Figure 9.

Figure 9: Site layout

The anemometer was mounted on a 17m tower. The pressure sensors communicated through a wifi mesh network to a base station that simultaneously records pressure data, wifi network statistics, and, from the reference anemometer, wind speed and direction. This is a small representation of the envisioned commercial setup with anemometers at 50m or higher representing turbine hub height and more sensors.

Data was collected for over nine weeks. Six weeks of data are useful for analysis with the other 3 weeks lacking data due mainly to low battery power at the nodes. Data was collected in large csv files and imported to a database through a data quality check procedure.

Data Handling

Data Cleanup

As first generation equipment running on batteries, the nodes would periodically lose battery power and therefore stop participating in the wireless mesh. As a consequence, the mesh would need to re-route signals sometimes resulting in data loss. We also saved time and money on database systems and wrote all the data to .csv files. While effective for most of the data collection, when sensors or the mesh would come and go, data in the .csv files could be duplicated or omitted. As a result, some data cleanup was required prior to working with the ML analysis.

Another important lesson learned from our data cleanup related to the anemometer. While our custom-built pressure sensor nodes performed well and collected useful data, we found that our low-cost off-the-shelf anemometer was too basic a model for our purposes, and our wind speed data was often aliased. To better approximate wind farm-caliber anemometers, we will need to upgrade our own anemometers and pursue redundant wind speed and direction readings in future pilot deployments.

Data

One second (1 Hz) data from all 19 sensors and the anemometer was successfully logged for the six weeks. This amounts to 3,628,800 seconds of data for all 20 data sources, totalling 72,576,000 data lines. Each line included time stamp, p, delta p, T, sensor health, etc..

An excerpt of processed input data is shown in the figure below. This is representative of variability in the anemometer data as well as tracking of delta p and delta *v*. We had only one sensor that didn't seem to exhibit a correlation between p and *v*,

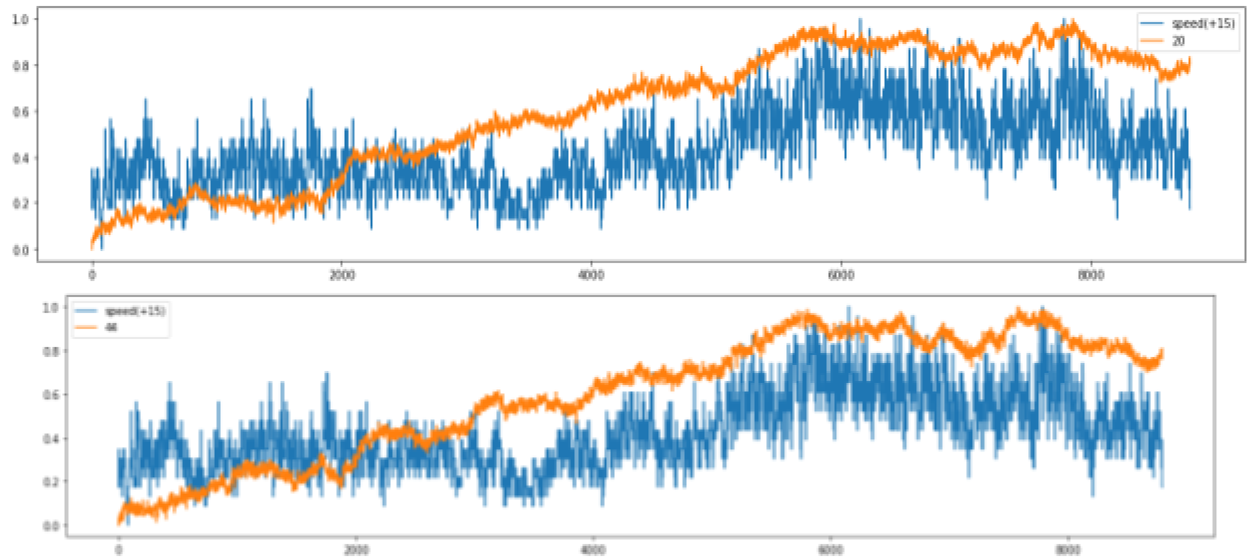

Figure 10: Exploratory Data Analysis

deltaP for one sensor, for one day
Blue = raw anemometer data change
Orange = raw pressure change

Analysis

The analyses presented in the results section below were produced by the team led by Deval Pandya of Impact2Rise, which is an ML consulting firm and advisor to Windscape.ai. The team deserves considerable credit for assessing multiple ML approaches, conducting the in depth analysis and providing clear summaries.

ML Modeling

We give a brief overview of the early analysis performed by the Impact2Rise ML analysis group. We shared our pilot data with Impact2Rise, but did not share our own analysis. This was done to ensure that Impact2Rise's analysis would not be influenced by our own results; that is, to ensure that their analysis served as an independent verification of our system.

After some exploratory data analysis in which they concluded there were no obvious correlations in the data, they decided on an initial ML model consisting of two decision tree-based models. A tree-based model, being nonparametric, is a logical choice for an easy-to-tune first pass model for prediction on data that exhibits no obvious trends. This is particularly true for our situation, since our hypothesis means that any useful predictive trends would necessarily be created by a complex physical system involving fluid flow over a large study area.

The prediction objective they chose for this early-stage analysis was to predict the anemometer wind speed 15 seconds ahead using current-time pressure data. Note that this early-stage analysis simplifies the full problem by turning it into a fixed-window prediction problem, rather than using proper time series methods. This is a suitable simplification for a first-pass analysis since time series methods are generally much harder to implement correctly, and our basic objective is to demonstrate the ability to predict anemometer speed.

The particular model they developed is schematically presented in Figure 11 below. As mentioned, it consists of two parallel decision tree models, whose predictions are combined to create the final prediction. The two models are an XGBoost model and a LightGBM model. These are two modern, highly-performant gradient boosted decision-tree models.

Feature engineering was also performed to create more informative features consisting of combinations of the raw per-sensor readings. This process serves a similar purpose for shallow learners like decision trees as the early layers would serve for a deep neural network. They used two automatic feature discovery algorithms: 1) a Truncated SVD algorithm that creates features by forming many new data matrices using sampled subsets of the data, taking the SVD (singular value decomposition) of these new matrices, then using the components of this SVD as new features, and 2) an “Interactions” algorithm, which tests simple addition, subtraction, multiplication, and division operations between two columns as potential features. A plot showing the relative importance of the five most important features, both raw (i.e., the original sensor readings) and engineered, is shown in Figure 12. The feature engineering process resulted in 104 new engineered features, which were used in combination with 14 original features as prediction inputs.

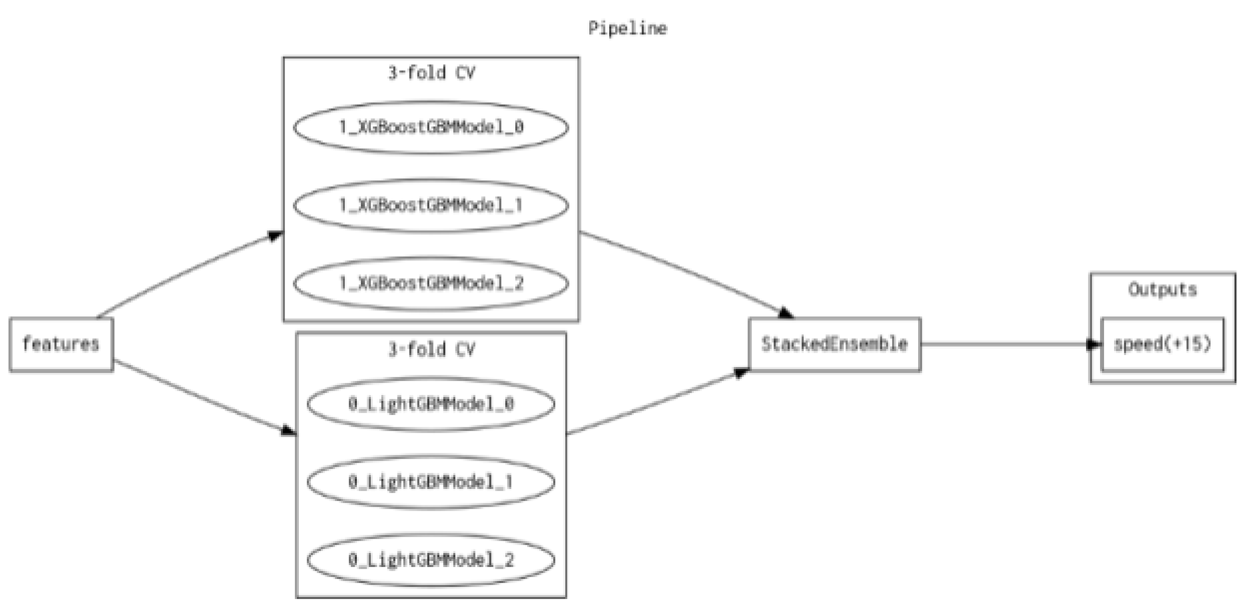

Figure 11: Ensemble Decision Tree Pipeline

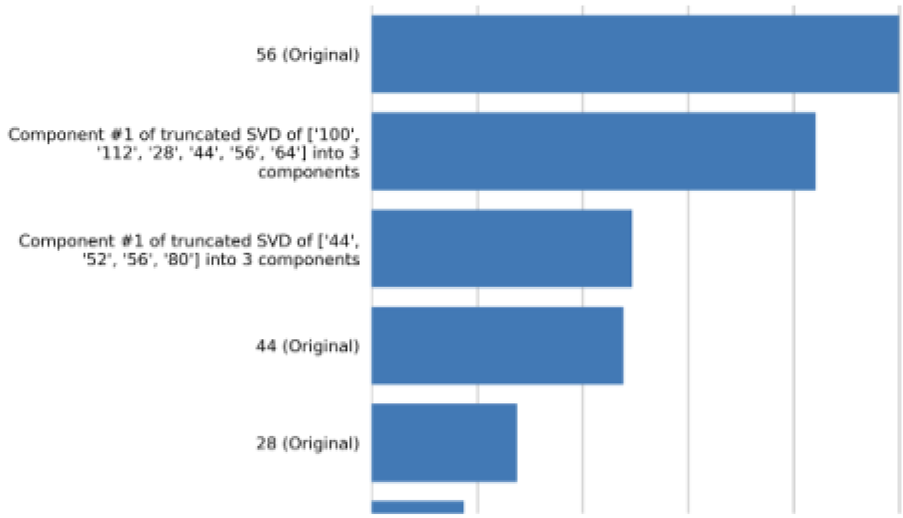

Figure 12: Relative importance of raw and engineered features

Finally, standard cross validation was performed to robustify the final model. This is a procedure that involves re-fitting the model on varying subsets of the data, and validation on the left-out subsets of the data, to guard against potential overfitting.

This final model performed very well. Figures 13 and 14 below show some illustrative examples of the prediction results. Figure 13 shows a simple scatter plot of predicted vs actual 15-second-ahead wind speeds. Linear regression on this correlation gives an R^2 of approximately 0.76.

Figure 13: Scatter plot of predicted wind speeds vs actual wind speeds with an R2 of approximately 0.76.

Figure 14 shows a timeseries plot of 15-second-ahead wind speed predictions overlaid on the actual wind speed for two hours in the early morning of July 22nd. Although the results are not perfect, the statistical model manages to predict all of the overall trends.

Two disadvantages of this tree-based model are that 1) the models have difficulty generalizing to distributional shift, which in our case seems to mean that a model trained on early-morning hours will not perform well in midday; and 2) as mentioned, this model takes a non-timeseries approach, meaning it cannot be used to predict ahead for any period of time other than 15 seconds. Both of these problems can be resolved by turning to a more powerful, timeseries-based approach such as an LSTM neural network. This approach has been the bulk of our own research effort so far.

Figure 14: Timeseries plot of 15-second-ahead predicted wind speeds (orange) vs true wind speed (blue).

Impact2Rise also reported some of their early-stage progress on attempting to train an LSTM deep neural network on our data. An example prediction timeseries is shown in Figure 15. This model again captures the overall trends but has not learned the smaller-scale fluctuations yet. Note that this LSTM result is a very early-stage result, before any heavy neural network engineering and tuning.

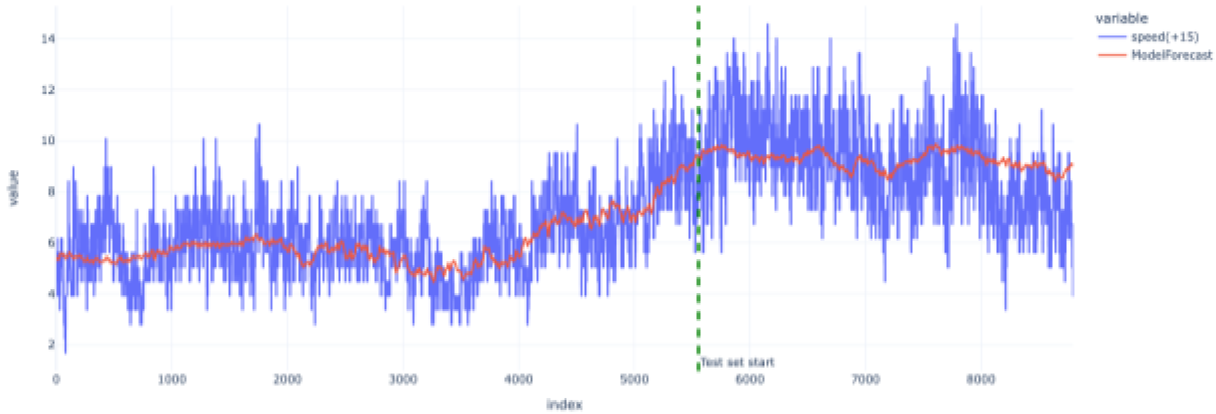

Figure 15: Timeseries plot of wind speed predictions from an early-stage LSTM neural network.

Conclusions

We are showing a close correlation between predicted wind speeds and actual wind speeds.

This is a first pass model that is not highly optimized, and a lot of hyper parameter tuning is needed to fine tune the results. Nonetheless, the model is now minutely capturing the direction of wind speed changes.

The model does very well on the training data and the test data within the sample data window. It does not generalize well for out of sample data which should be helped by more modeling experience and more physics-informed engineered features.

Additional work is needed to expand this analysis to generalize over seasons and to perform for multiple wind turbine reference points. Our upcoming pilots at operating wind plants will have winds and reference points to pursue this.

A more sophisticated, powerful model such as a deep LSTM neural network will likely further improve our prediction capabilities. Building this type of model is established data science and is being pursued in our ongoing analysis work.

