Machine Learning Algorithms for Wind Turbine Performance Monitoring
Tracking optimal yaw alignment based on SCADA data

PO.168

Sebastian Kaus, Bert Gollnick, Niko Mittelmeier
Senvion

SENVION
wind energy solutions

Abstract

Stakeholders in wind energy are interested in optimal wind farm operation. Key to an optimal wind farm performance is high availability and good quality of power performance during operation.
Wind turbines provide a large amount of data which is regularly used for availability calculation and ad-hoc analysis.

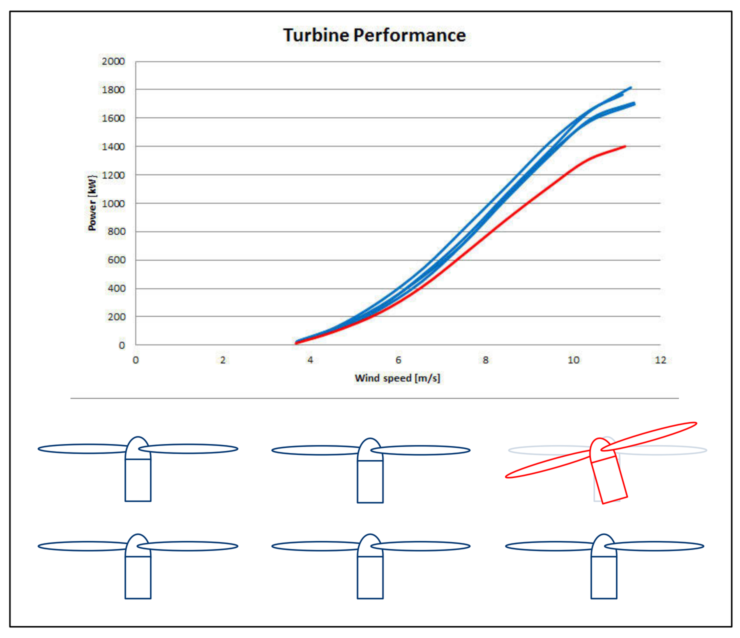


Figure 1. Misaligned turbine and effects on power curve

A wind farm monitoring method based on an ensemble of machine learning technics has been developed. The algorithms are trained on SCADA data from turbines with known properties. For example, in a test wind farm, one turbine has been forced to operate with an yaw error for a short period and the algorithms challenged to identify it. The ensemble of algorithms correctly identified the misalignment and the varying performance of the different machine learning techniques could be observed.

Objectives

Senvion analyses turbines for their performance with special respect to yaw misalignment. An approach using machine learning algorithms to classify analyzed turbines opens up new ways to detect underperformance and elaborate automated solutions to improve the yield of the turbine. Close interaction and standardized interfaces between data acquisition of the turbine in the field, the data analytics team and the service and maintenance department enables Senvion to quickly act on identified performance issues. A growing data base and feedback from the field about analysis results ensures the improvement and quality of the analysis.

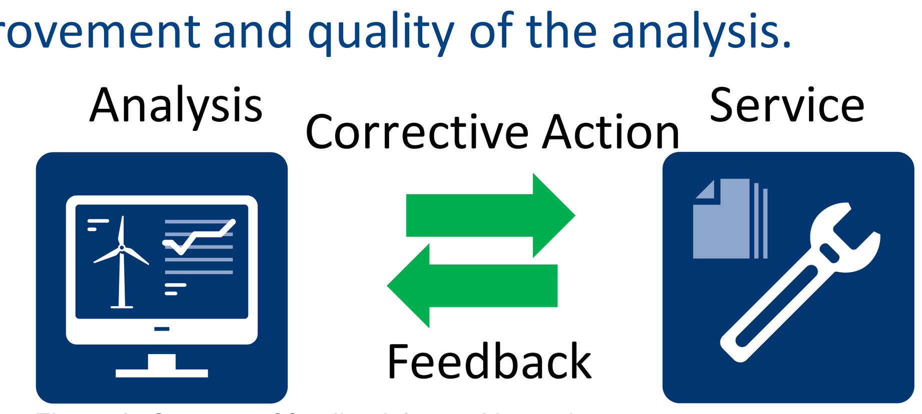


Figure 3. Concept of feedback loop with service

Methods

The opportunities for Senvion to run specific analysis on their manufactured turbines are manifold. Advantages of Senvion's approach are the large data base, knowledge about technical details and access to parameter sets and turbine master data. Turbines that are serviced by Senvion are also a valuable source of feedback and provide essential information about the validity.

An ensemble of different machine learning algorithms is used to categorize turbines and identify misalignment in the fleet.

To support the analysis, calculated Key Performance Indicators (KPIs) are used instead of raw SCADA data. These KPIs are normalized values with special respect to meaningful information about yaw misalignment:

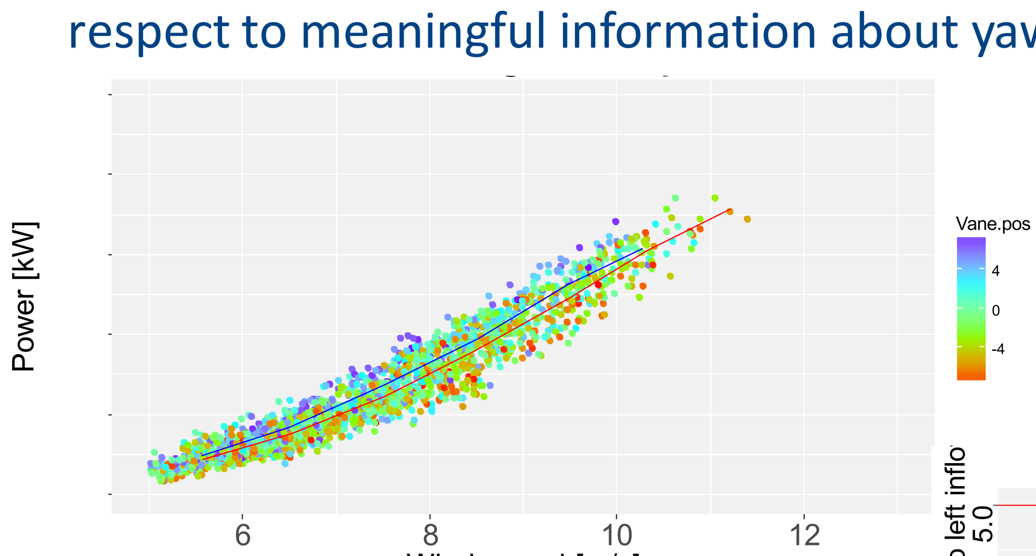


Figure 4. Visualisation of KPI generated from power scatter plot

Scatter plot of power as function of the measured wind speed and inflow sector. Difference between averaged sectors results (lines) serves as KPI.

The AEP ratio is calculated by inflow sectors. The difference between the sectors serves as KPI

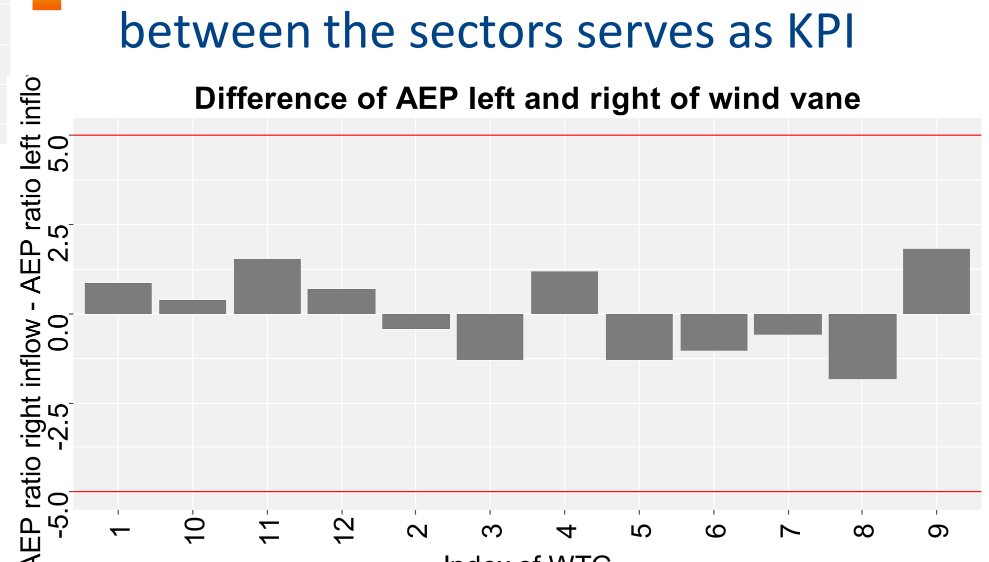
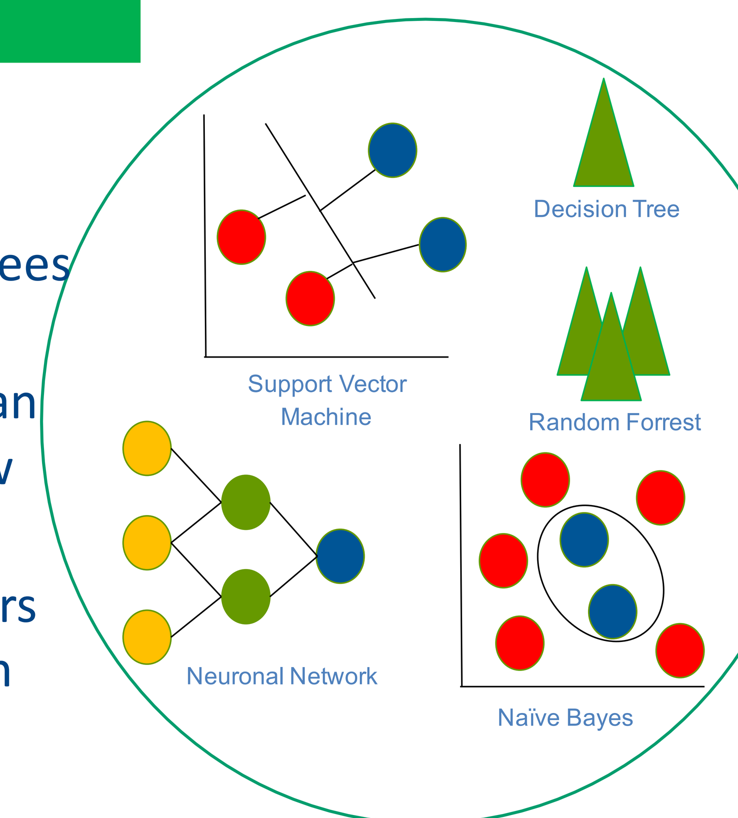


Figure 5. Visualisation of KPI generated from AEP ratio

Ensemble of Algorithms

- Decision Tree: Classifies input data according to larger-or-smaller-as-rules
- Random Forest: A "forest" of decision trees with variable start values
- Neuronal Network: Resembles the human brain and makes decisions based on how often a synapsis is used
- Support Vector Machines: Creates vectors and adds them to values to classify them with a hyperplane
- Naïve Bayes: Classifies by attributes of values



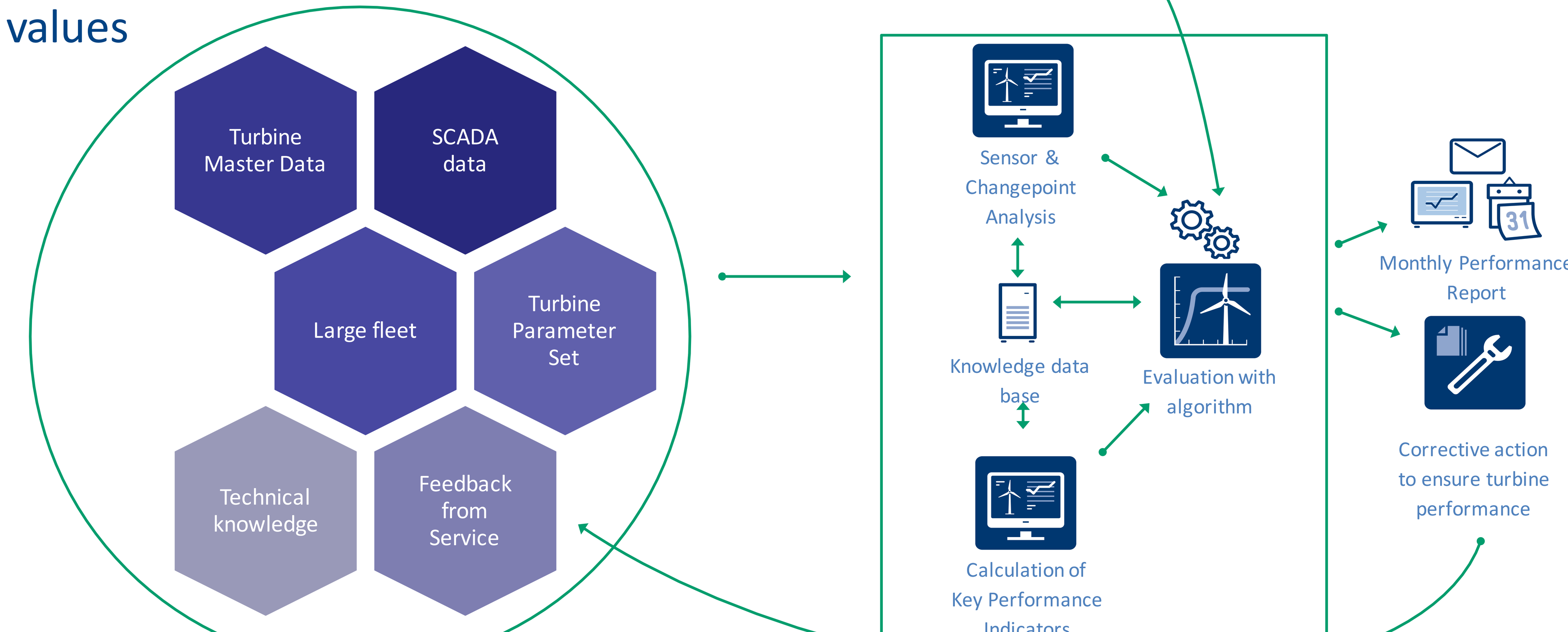


Figure 6. Flowchart of Performance Monitoring with machine learning algorithms

Results

	Bayes	TREE	FORREST	SVM	NNET	Sum	combined	Turbine
1	0	0	0	1	0	1	No	2
2	0	0	0	1	0	1	No	9
3	0	0	0	1	0	1	No	8
4	0	0	0	1	0	1	No	1
5	0	0	0	1	0	1	No	10
6	0	0	0	1	0	1	No	4
7	0	0	0	1	0	1	No	5
8	0	0	0	1	0	1	No	6
9	0	0	0	1	0	1	No	3
10	0	0	0	1	0	1	No	7
11	0	0	0	1	0	1	No	12
12	0	0	0	1	0	1	No	11

Figure 7. Analysis results before wind vane offset

	Bayes	TREE	FORREST	SVM	NNET	Sum	combined	Turbine
1	0	0	0	1	0	1	No	2
2	0	0	0	1	0	1	No	9
3	1	0	1	1	1	4	Yes	8
4	0	0	0	1	0	1	No	1
5	0	0	0	1	0	1	No	10
6	0	0	0	1	0	1	No	4
7	0	0	0	1	0	1	No	5
8	0	0	0	1	0	1	No	6
9	0	0	0	1	0	1	No	3
10	0	0	0	1	0	1	No	7
11	0	0	0	1	0	1	No	12
12	0	0	0	1	0	1	No	11

Figure 8. Analysis results during wind vane offset

	Bayes	TREE	FORREST	SVM	NNET	Sum	combined	Turbine
1	0	0	0	1	0	1	No	2
2	0	0	1	1	0	2	No	9
3	0	0	0	1	0	1	No	8
4	0	0	0	1	0	1	No	1
5	0	0	0	1	0	1	No	10
6	0	0	0	1	0	1	No	4
7	0	0	1	1	0	2	No	5
8	0	0	0	1	0	1	No	6
9	0	0	0	1	0	1	No	3
10	0	0	1	1	0	2	No	7
11	0	0	0	1	0	1	No	12
12	0	0	0	1	0	1	No	11

Figure 9. Analysis results after wind vane offset

In a wind farm of 12 turbines (MM92) with a nominal power of 2kW, one turbine was forced to operate with 5 degrees misalignment. The algorithm was able to successfully detect the misalignment. In the periods before and after the test, the algorithm shows no misalignment. The training set consists of eight wind farms with a total of 1100 data samples.

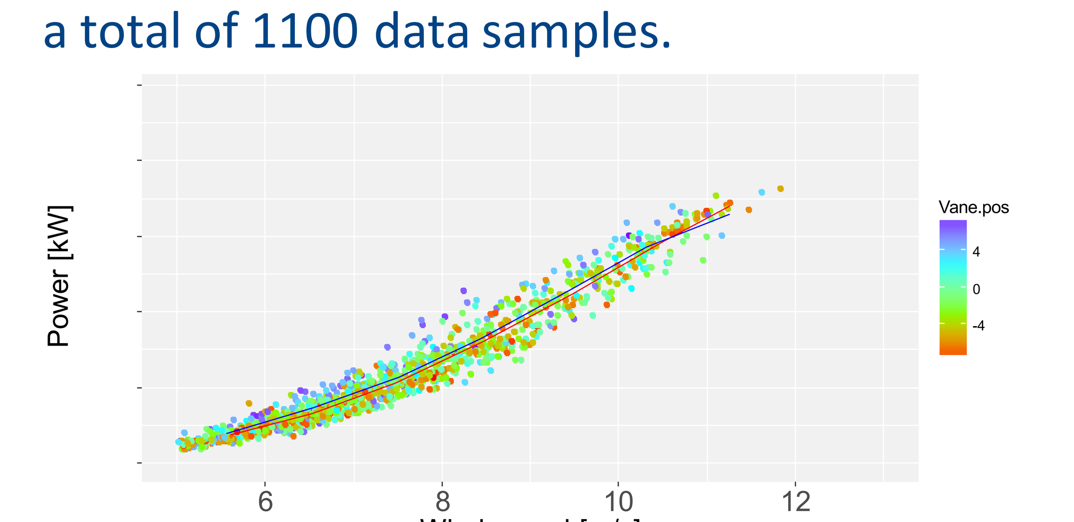


Figure 10. Visualized KPI from misaligned turbine. Compare with Figure 4. to see difference to a properly aligned turbine

Conclusions

To obtain good and reliable results, a deep understanding of turbine technology is required for modelling. Machine learning helps to create more sophisticated models. These algorithms create knowledge based on experience and permanently update this knowledge. In general, the more data available and taken into account, the better the model and results. By knowing a turbines' ideal positioning, the OEM is able to improve its overall fleet performance by applying machine learning techniques.
Machine learning aggregates different statistical approaches for supervised and unsupervised learning based on large datasets. Similar to predictive maintenance, we show that machine learning can also be applied to maintain optimal turbine performance.

References

1. Breiman, L. (2001), Random Forests, Machine Learning 45(1), 5-32.
2. Breiman L., Friedman J. H., Olshen R. A., and Stone, (1984), C. J. Classification and Regression Trees. Wadsworth.
3. Ripley, B. D. (1996) Pattern Recognition and Neural Networks. Cambridge.
4. B. Schölkopf, A. Smola, R. Williamson, and P. L. Bartlett. (2000) New support vector algorithms. Neural Computation, 12, 1207-1245