

PUBLIC

Westerduinweg 3
1755 LE Petten
P.O. Box 15
1755 ZG Petten
The Netherlands

www.tno.nl

T +31 88 866 50 65

TNO report

TNO 2020 R11206 | Final report

**Investigation into automating wind turbine
underperformance detection using social statistics**

Date	7 September 2020
Author(s)	Clym Stock-Williams Dennis Wouters Maciej Maj
Copy no.	1
No. of copies	1
Number of pages	39
Number of appendices	0
Customer	Dutch Ministry of Economic Affairs and Climate Policy (EZK)
Project name	Automated Underperformance Detection for Wind Turbines
Project number	060.38599

All rights reserved.

No part of this publication may be reproduced and/or published by print, photoprint, microfilm or any other means without the previous written consent of TNO.

In case this report was drafted on instructions, the rights and obligations of contracting parties are subject to either the General Terms and Conditions for commissions to TNO, or the relevant agreement concluded between the contracting parties. Submitting the report for inspection to parties who have a direct interest is permitted.

© 2020 TNO

PUBLIC

Executive Summary

This report describes the work performed as part of the PPP Toeslag project “Automated Underperformance Detection for Wind Turbines”.

The project aimed to take a novel approach to wind farm condition monitoring and automated alerting, based on Bayesian statistics and the relative behaviour of wind turbines, with the following goals:

- To be able to automatically detect and quantify underperformance in a wind farm.
- To do this as much as possible using only 10 minute SCADA data, which is cheap and readily available (although often unreliable).
- To assess the additional value of a minimal set of data from additional measurements (e.g. a Lidar campaign) – in order to improve information sufficiently to take maintenance decisions.

For this purpose, Fortum provided TNO with several years of commercially-sensitive SCADA data from one of their onshore wind farms in northern Europe, along with data from expensive nacelle-mounted Lidar campaigns which established yaw misalignment on three turbines.

The majority of analyses performed on SCADA data in both academic and industrial contexts involves one form or another of individual turbine analysis, in particular of turbine power against nacelle anemometer wind speed. The difficulty with this—which is very widely understood in the industry—is that, even with the built-in calibration to account for the effect of the rotor on the wind, its measurements are not accurate enough to support reliable detection of underperformance.

This report—by contrast—elaborates a concept of “social statistics”, where turbines compare their behaviour to each other, enabling inference of their slowly-varying performance characteristics, freed of the many uncertainties that arise from unmeasured or poorly-measured wind conditions. Properties of the turbines are inferred based on information from the individual turbine, pair, or wind farm level (depending on the complexity of information required to perform the inference) as illustrated in Figure 1.

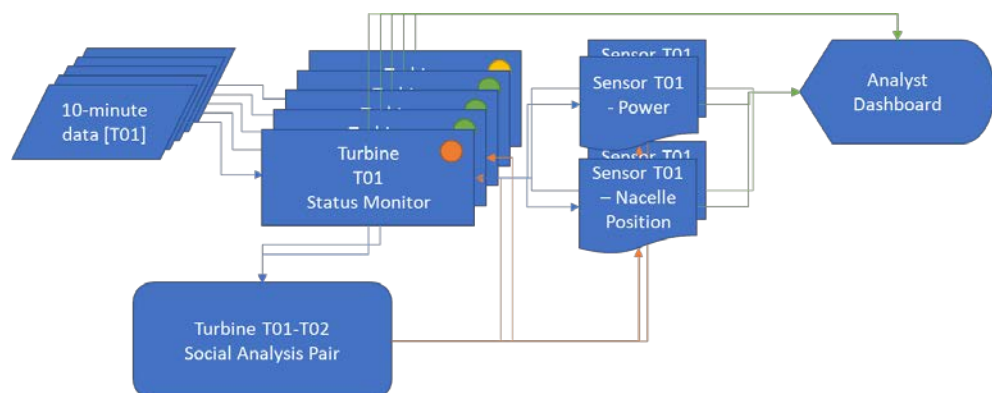


Figure 1 Overview of social statistics framework.

Algorithms are here provided for important parts of the framework, such as the automatic creation of valuable turbine pairs. Further, a robust software architecture has been designed and tested, enabling the required event-driven processing of the 10-minute average SCADA, and allowing analysts to implement additional inference logic for different degradation modes or turbine / sensor statuses.

Due to the particular wind farm data set provided, which included a rather valuable measurement campaign of nacelle Lidar data, the decision was taken early on to focus exclusively on implementing an algorithm for yaw misalignment detection.

However, one of the key learnings from this project is that nacelle yaw sensors can be highly unreliable. It is proven here that separation of yaw misalignment from nacelle yaw direction miscalibration is impossible without any source of independent truth to robustly re-calibrate the true nacelle direction.

To provide a result of value to the industry, an automated method for using measured differences in yaw direction between turbine pairs to infer the sum of yaw misalignment and miscalibration has been developed, using Bayesian inference. This is shown with a toy example to be able to estimate correctly each individual turbine's yaw error within a fraction of a degree. This is industrially useful as it can identify turbines for maintenance, and/or be used automatically to remove the calibration error.

This method can include sources of truth such as a nacelle-mounted Lidar campaign to estimate yaw errors (due to the combination of miscalibration and misalignment). An example is shown in Figure 2 for the results inferred for the test wind farm.

One way to estimate the accuracy of the method is to compare the prediction of yaw misalignment (assuming zero miscalibration) on turbines which were later subjected to a Lidar campaign. The predicted values were within $[-1.5, +2.8]^\circ$ of the measured values. Given the length of time elapsed between prediction and measurement, the regularity of yaw sensor de-calibrations, and some concerns about the results from the lidar measurements themselves, this seems reasonable.

It is likely that the methods outlined here will be more successful on wind farms where the effect of orography is either known, or less pronounced. Sites with a higher spread of incoming wind directions should also be more fruitful. However, the wind farm does not need to have many turbines for social statistics to be effective, so long as the turbines are not too far apart both horizontally and vertically.

Future extensions to this method should place more emphasis on detection of other degradation modes, such as blade damage. This can be achieved by looking more deeply at combining the power ratios of the turbine pairs with the ratios and differences of other sensors.

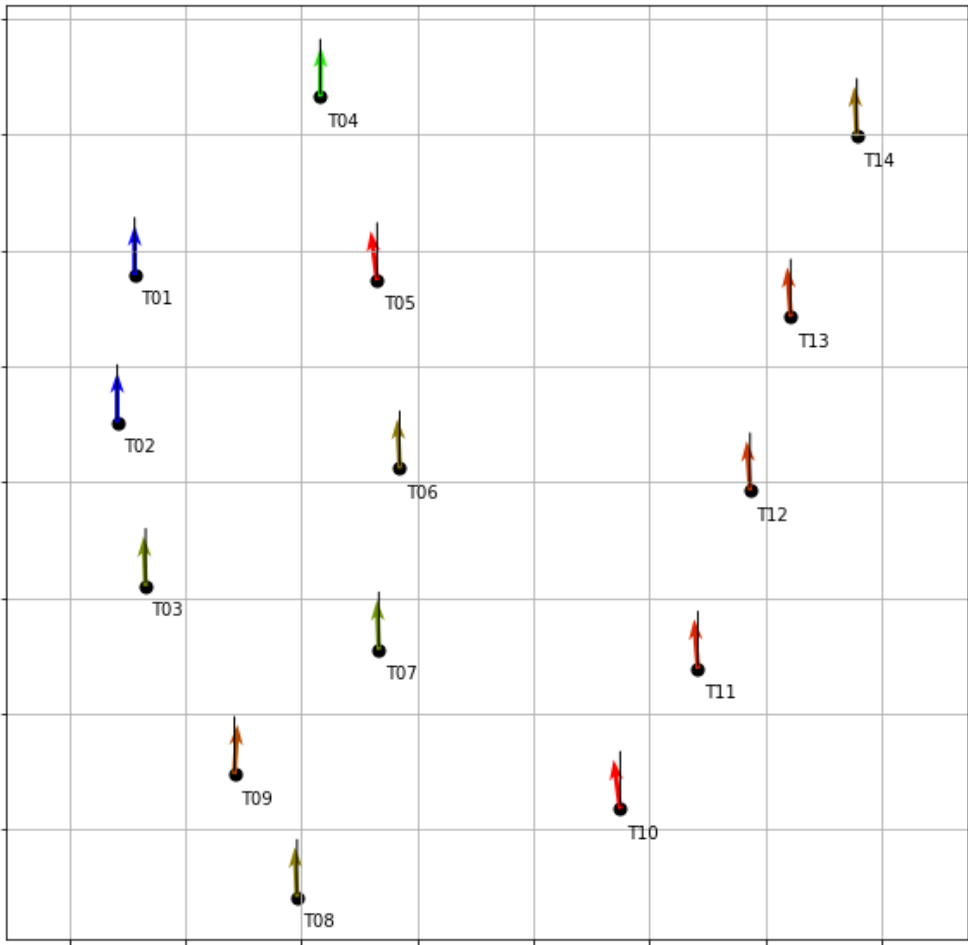


Figure 2 Illustrative dashboard of inferred yaw misalignments for the wind farm used in this study.

Contents

Executive Summary	2
1 Project Aims and Report Structure.....	6
2 Wind farm data set.....	7
2.1 Wind farm description	7
2.2 SCADA Data.....	7
2.3 WRF Meteorological Data	12
3 Social Statistics Methodology.....	15
3.1 Motivation and Literature Review	15
3.2 Social Analysis Framework	17
3.2.1 Independent turbine inference	19
3.2.2 Turbine pair inference.....	20
3.2.3 Wind farm inference.....	21
3.3 Software Structure	22
3.4 Yaw Analysis	25
4 Results and Analysis.....	29
4.1 Data preparation	29
4.2 Lidar data analysis.....	29
4.2.1 Calibration error	29
4.2.2 Yaw misalignment.....	30
4.3 Applying social statistics	34
5 Conclusions and Recommendations	38

1 Project Aims and Report Structure

This report details the work undertaken during the PPP Toeslag project “Automated Underperformance Detection for Wind Turbines”. The main aim of this project was to investigate the value in taking a novel approach to wind farm condition monitoring and automated alerting, based on Bayesian statistics and the relative behaviour of wind turbines. The approach was supported by Fortum, who provided data from an onshore wind farm in northern Europe, where yaw misalignment in three turbines had been detected using nacelle-mounted Lidar campaigns.

The objectives of the project were:

- To be able to automatically detect and quantify underperformance in a wind farm.
- To do this as much as possible using only 10 minute SCADA data, which is cheap and readily available (although often unreliable).
- To assess the additional value of a minimal set of data from additional measurements (e.g. a Lidar campaign) – in order to improve information sufficiently to take maintenance decisions.

For this purpose, Fortum provided several years of commercially-sensitive SCADA data from one of their wind farms, along with expensive nacelle-mounted Lidar data sets and previous analysis. These data sets are described in section 2.

Next, the methodologies applied are described in section 3. These are built to be implementable online in a SCADA monitoring system.

In section 4, these methodologies are applied to the data set from Fortum, in order to assess their effectiveness, in particular their ability to predict the yaw misalignment already seen. Additional benefits and difficulties are discussed and demonstrated.

Finally, section 5 summarises the conclusions from this study, the potential next steps for promising lines of research, and recommendations to both Fortum and the wider wind energy industry.

2 Wind farm data set

2.1 Wind farm description

Wind farm data were provided by Fortum for an onshore wind farm in Northern Europe. The turbines are located as shown in Figure 1.

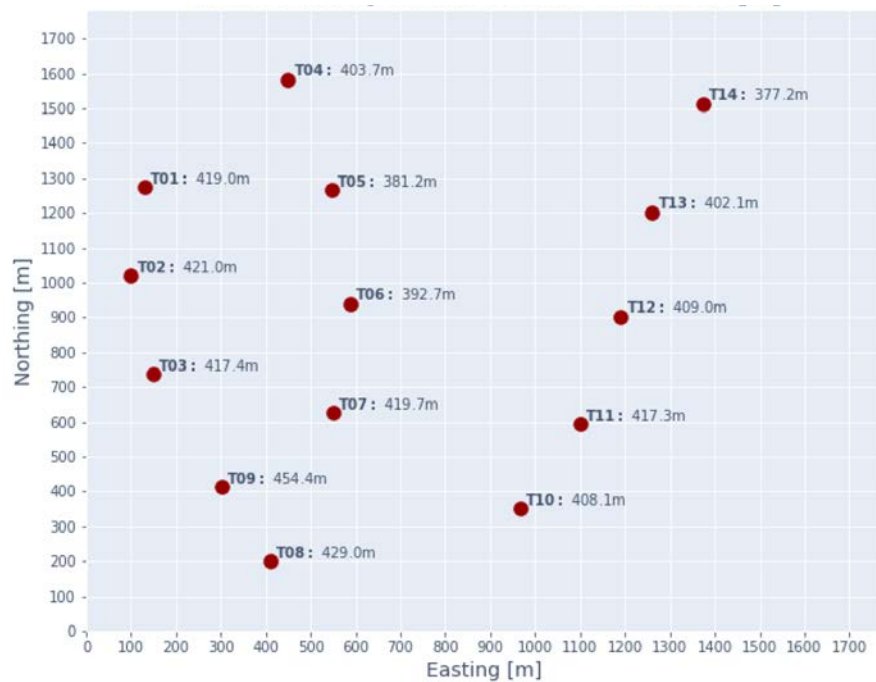


Figure 3 Turbine heights above mean sea level, and locations (in UTM relative to an arbitrary origin).

The wind farm consists of 14 turbines, less than 10 years old, operating across an area of 1300m by 1400m, with a maximum elevation difference of 80m.

The terrain is expected to have a significant effect on the wind speeds and directions across the site during normal operation.

2.2 SCADA Data

Supervisory Control and Data Acquisition (SCADA) data were provided for a period of more than 7 years, with a 10-minute frequency. While many hundreds of parameters were available in the manufacturer's documentation, and more than 70 parameters were provided in the data set, most of these were missing or corrupted. As one example, wind direction measurements from a nacelle wind vane (or sonic anemometer) were not present, even though they must be available for the control system to function. The parameters used in this study are summarised in Table 1.

The data were loaded into Python, structured into a *pandas DataFrame*, and then stored in *pickle* files for further use in the project. An illustration of the cleaned data is shown in Figure 2.

Table 1 SCADA parameters used

Parameter description	Comments
Ambient Temperature (°C)	Mean, min, max, standard deviation
Active Power (kW)	Mean, min, max, standard deviation
Nacelle Position (°)	Mean, min, max, standard deviation
Main Shaft Speed (RPM)	Mean, min, max, standard deviation
Wind Speed Active Sensor (m/s)	Mean, min, max, standard deviation
Duration Turbine Released to Operation	Between 0 and 600 seconds
Duration Turbine Operating	Between 0 and 600 seconds
Duration Wind Speed between Cut-in and Cut-out	Between 0 and 600 seconds

Figure 4 Illustration of cleaned DataFrame with SCADA data

wtc_ActPower_mean					
<i>idx:</i> timestamp	T01	T02	...	T14	
2012-09-20 00:40	0.0	0.0	...	0.0	
2012-09-20 00:50	0.0	0.0	...	0.0	
2012-09-20 01:00	0.0	0.0	...	0.0	
...	...	...	...	...	

Outlier detection and removal is an essential first step before performing analysis. Several methods for detecting outliers are standard in the statistics and data science literature. Here, we illustrate the process used by focusing on identifying spikes in nacelle anemometer-measured wind speed (which seemed to occur most frequently when the turbine is turned on again after maintenance).

Two methods were applied to flag suspicious data points, for subsequent manual review before removal or retention:

- 1 Considering only a single variable (each turbine's wind speed data in this case), and flagging all times where the value exceeds the 99th percentile. This non-parametric method was chosen over alternative such as the z-score, because the data are not Normally-distributed (even when taking the logarithm);
- 2 Calculating the absolute difference of the measured variable from its equivalent in a reference meteorological data set (described in section 2.3), then flagging values greater than the 90th percentile (or some manually-chosen threshold selected *a priori* based on the particular parameter, e.g. 5m/s for wind speed). With this approach it is appropriate to be more aggressive at flagging data, since the use of a model means that the physical processes explaining some extreme values will be included, and therefore deviations should only occur when data errors are present.

The results of the outlier removal process are shown using boxplots. These diagrams are a powerful tool for visualising the distribution of data. An explanatory example is shown in Figure 3, and the before-and-after plots for each turbine's measured wind speed are given in Figure 4.

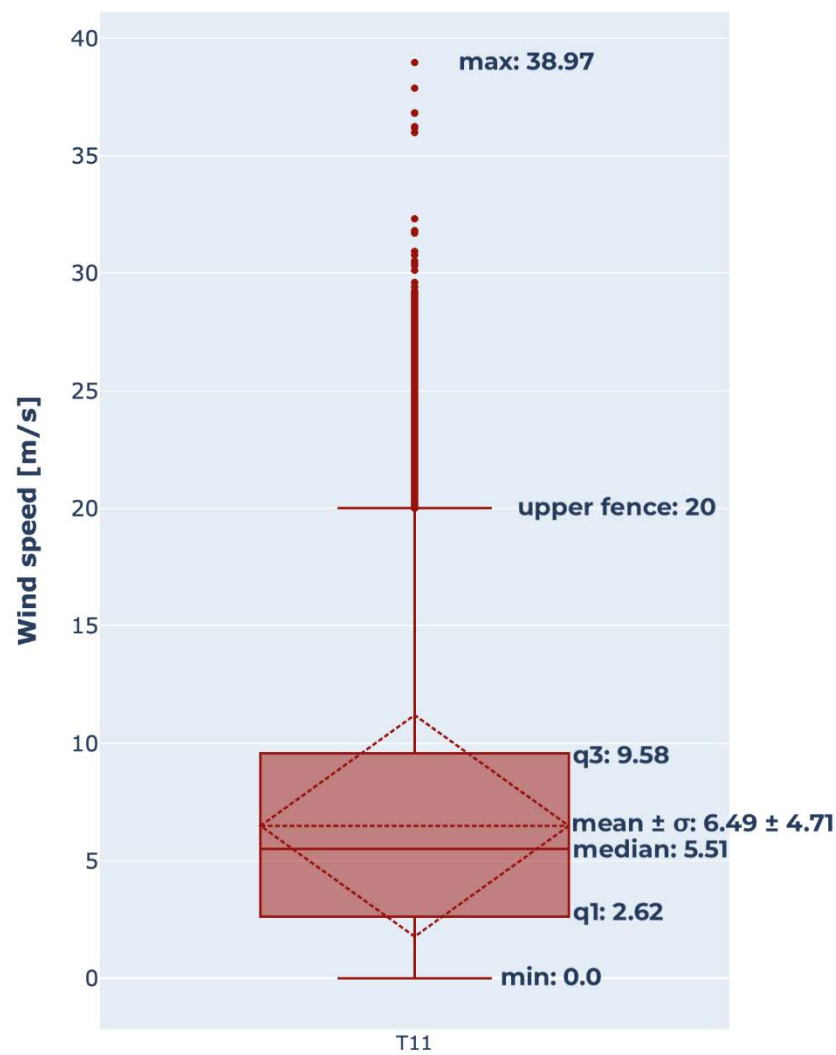


Figure 5 Illustrative boxplot for nacelle anemometer-measured wind speed. The solid lines indicate non-parametric values (from bottom: lower fence, lower quartile, median, upper quartile, upper fence). The fences are set at 1.5 times the inter-quartile range, or the minimum/maximum value, whichever is nearer the median. The dotted horizontal line shows the mean and the extremes of the dotted triangles shown one standard deviation away from this mean. All points lying outside the fences are plotted as dots.

Further examination of the time series of nacelle position was undertaken, given its importance in determining the possible yaw misalignment, and the lack of wind vane data.

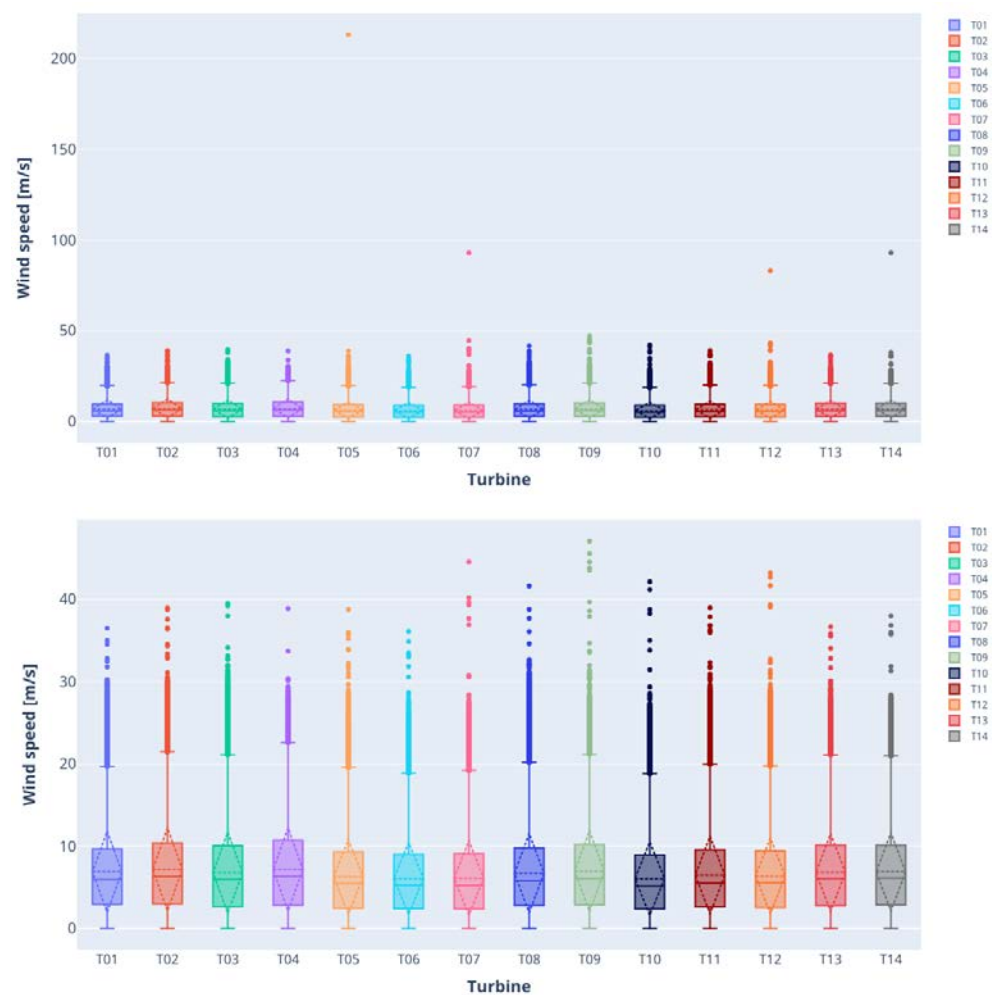


Figure 6 Illustration of outlier removal on SCADA wind speed data, using boxplots described in Figure 3. Top: before cleaning; bottom: after cleaning.

Figure 5 illustrates how regular large shifts in measured yaw position occur. In the top graph, there is a divergence in measurement of wind of up to 180 degrees. The bottom graph, in a period almost 1 year later, demonstrates that the miscalibration of the yaw position sensors has been largely corrected, and the divergence is now around 30 degrees. However, the relative position of the measurements exhibits systematic differences after each direction shift. For instance, following the purple line for turbine T02, after February 12th it has the most northerly yaw position measurement. The next time the westerly winds occur, just before the mark for February 19th, it is the most southerly. This relative position persists for the easterly wind period following, before reverting to being the most northerly.

Such changes could be attributed to a combination of: an error in the sensor; a change in sensor measurement due to external influence; a true change in the controller resulting in misalignment; or a systematic change in inflow wind in time.

Some of these proposals are less plausible than others, and can be discounted after enough data has been collected. For example, any change in time attributed to orography should be fully explained by, and consistent with, the change in direction.

Sensor hysteresis after large yaw changes are possible, but have not been confirmed by any wind turbine manufacturer.

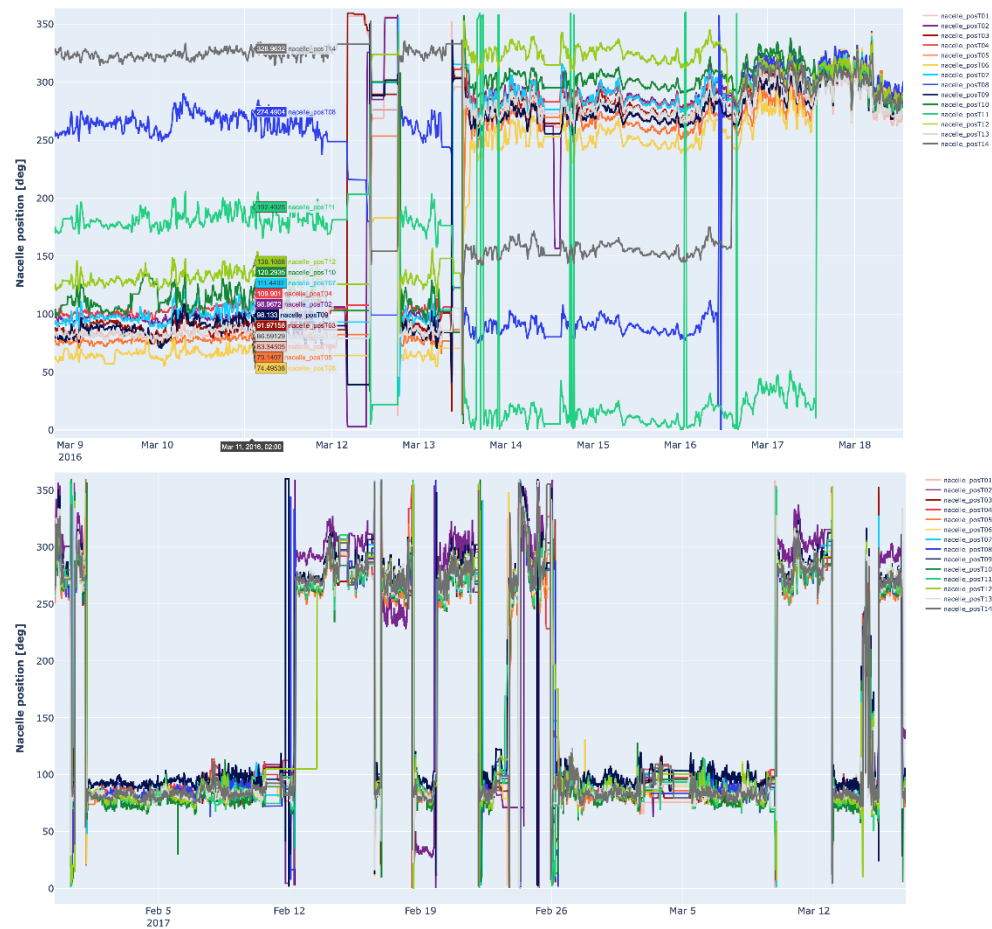


Figure 7 Illustration of nacelle position sensor calibration shifts. Top: a period with large disagreements between turbines; bottom: a period with small disagreements between turbines.

Finally, the contracted power curve, shown in normalised form in Figure 6, was provided.

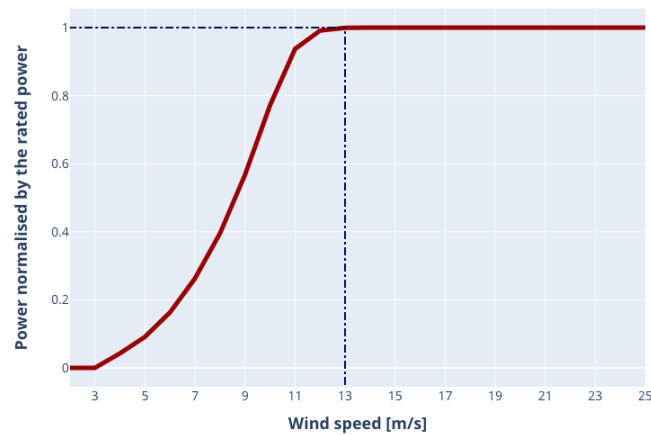


Figure 8 Turbine manufacturer's contractual power curve, with normalised power, and rated conditions indicated by the blue dash-dotted lines.

2.3 WRF Meteorological Data

Meteorological data at one-hour frequency were provided by Fortum from a well-known hindcast model provider. This was intended to replace the missing met mast data, and comprised the parameters shown in Table 2.

Table 2 Available WRF parameters

Parameter description
Wind Speed (m/s)
Wind Direction (°)
Atmospheric Pressure (Pa)
Ambient Temperature (K)
Relative humidity with respect to ice (%)
Relative humidity with respect to liquid water (%)

This data set enabled some exploratory data analysis to understand the conditions on site. First, the wind rose is shown in Figure 7. As can be seen, the wind is highly bi-directional, with most energetic winds coming from the East.

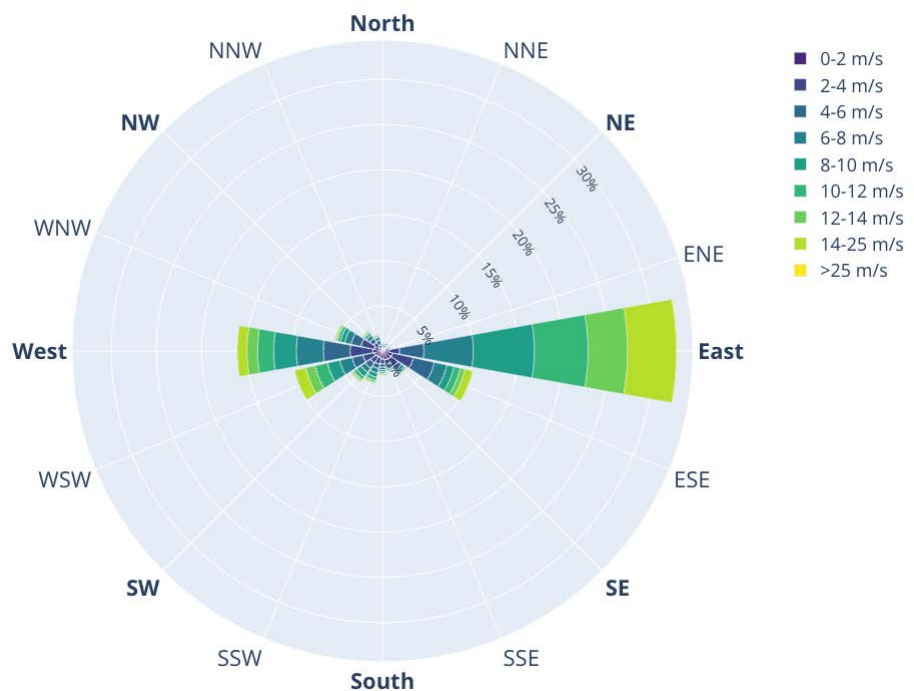


Figure 9 Wind rose created using WRF data for the wind farm under consideration

Next, a comparison of wind speeds and directions from the WRF versus turbine T04 is displayed in Figure 8. As can be seen, the agreement is poor (despite time zone shifting to align the series correctly). For the wind direction, this may be partly accounted for by the response time and control of the nacelle position versus the true wind direction. However, the wind speed comparisons were similar for every turbine, and inspection of time series plots shows that significant and persistent disagreements of up to 5m/s are present, indicating that the WRF data is not accurate in this location with complex terrain.

Due to this discrepancy, the WRF was not treated as a replacement for a meteorological mast.

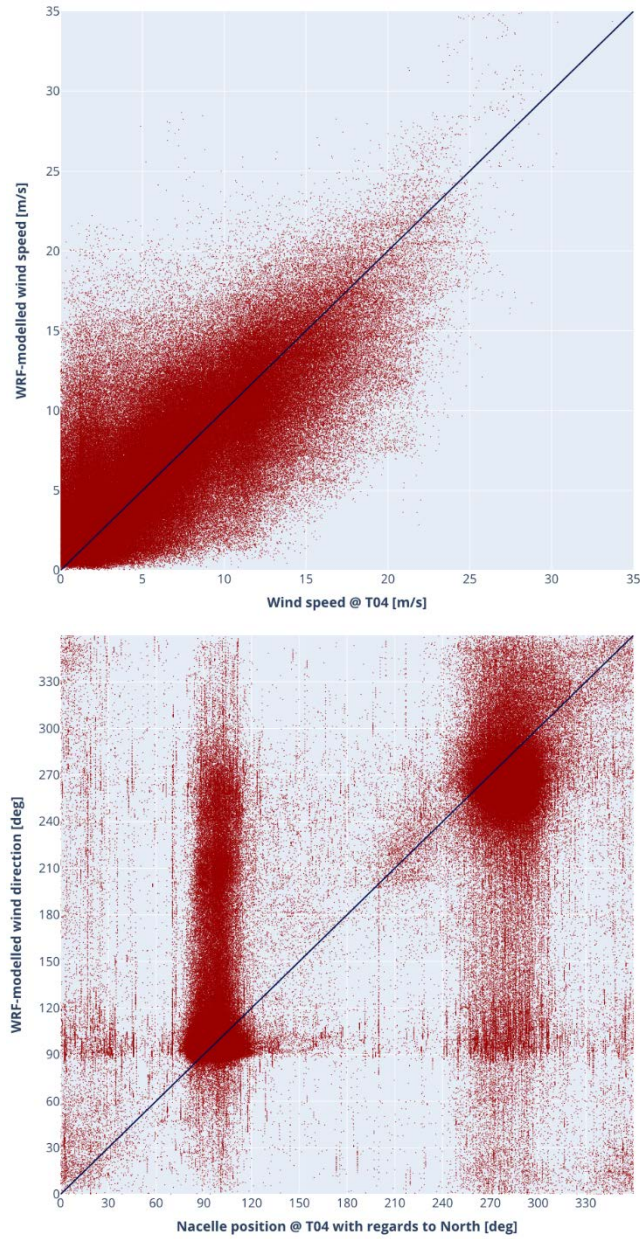


Figure 10 Comparison of WRF versus SCADA data. Top: wind speed from model and T04 nacelle anemometer; bottom: wind direction from model and nacelle position of T04.

Air density was nevertheless calculated as follows.

First, the reference ambient temperature T_{ref} was corrected to each turbine's height h , using the temperature lapse rate $L = 0.0065K/m$:

$$T_h = T_{ref} - L \cdot h$$

The vapour pressure of water p_v is then calculated as follows, where the relative humidity RH_{lw} is input as a %:

$$p_v = \left(6.108 \cdot 10^{\frac{7.5(T_h - 273.15)}{T_h - 273.15 + 237.8}} \right) \cdot RH_{lw}$$

The partial pressure of water vapour at each turbine's height $p_{h,vap}$ is then calculated, using the molar mass of water vapour $M_{vap} = 0.018 \text{ kg/mol}$ and the specific gas constant for water vapour $R_{vap} = 461.5$:

$$p_{h,vap} = p_v \cdot \left(1 - \frac{L \cdot h}{T_h} \right)^{\frac{g \cdot M_{vap}}{L \cdot R_{vap}}}$$

Next, the observed atmospheric pressure p_{ref} is corrected to the partial pressure of dry air at each turbine's height $p_{h,dry}$, using the molar mass of dry air $M_{dry} = 0.029 \text{ kg/mol}$ and the specific gas constant for dry air $R_{dry} = 287.1$:

$$p_{h,dry} = (p_{ref} - p_v) \cdot \left(1 - \frac{L \cdot h}{T_h} \right)^{\frac{g \cdot M_{dry}}{L \cdot R_{dry}}}$$

Finally, air density is calculated using:

$$\rho = \frac{p_{h,dry}}{R_{dry} \cdot T_h} + \frac{p_{h,vap}}{R_{vap} \cdot T_h}$$

3 Social Statistics Methodology

3.1 Motivation and Literature Review

Wind turbine SCADA data is analysed by wind farm owners in order to:

- 1) Summarise wind farm performance across a portfolio and enable asset management to focus on poor performers;
- 2) Identify poorly-performing turbines in a wind farm and prioritise them for corrective action;
- 3) Detect or predict damage or other problems with the turbine which require maintenance.

The focus of this project is on the second topic. The methodology outlined in section 3.2 and implementation in 3.3 are deliberately generic, giving a framework for the detection of underperformance and attempted attribution to any fault mode, where there is sufficient physical knowledge. In this project, there was a particular emphasis on detecting yaw misalignment, so section 3.4 focuses on that use case.

The majority of analyses performed on SCADA data in both academic and industrial contexts involves one form or another of individual turbine analysis, in particular of power curves¹. In this approach, poorly-performing turbines are identified by comparing a graph of power output against wind speed with the warranted power curve of the turbine.

The measurements of wind speed used for this analysis are usually from the anemometer (commonly a cup, sometimes sonic) placed on top of the wind turbine nacelle, behind the rotor. The difficulty with this—which is very widely understood in the industry—is that, even with the built-in calibration to account for the effect of the rotor on the wind, its measurements are not accurate enough to support reliable detection of underperformance². For this purpose the IEC 61400-12-1 standard has been created, which requires a separate measurement placed upstream of the wind turbine in undisturbed flow.

Even if a meteorological mast is available (which is often not the case, as for the wind farm used in this study) the wind field changes across the wind farm, largely due to wakes and orography (terrain). Thus this wind speed cannot be a suitable reference for all the turbines on a farm.

As an aside, one very useful diagnostic curve for individual turbines, which does not rely on uncertain wind speed measurements, is the torque-speed curve shown in Figure 9. When torque is not available in the SCADA system, it can be calculated from the active power P and shaft speed ω (in revolutions per minute, rpm) as follows:

$$\Gamma = \frac{30}{\pi} \cdot \frac{P}{\omega}$$

¹ Sohoni et al., “A Critical Review on Wind Turbine Power Curve Modelling Techniques and Their Applications in Wind Based Energy Systems”, J. Energy, 2016

² A. Albers, J. Mander and G. Gerdes, “Analysis of wind turbine control and performance based on time series data of the wind farm monitoring system.” Proceedings of EWEC2003, Madrid.

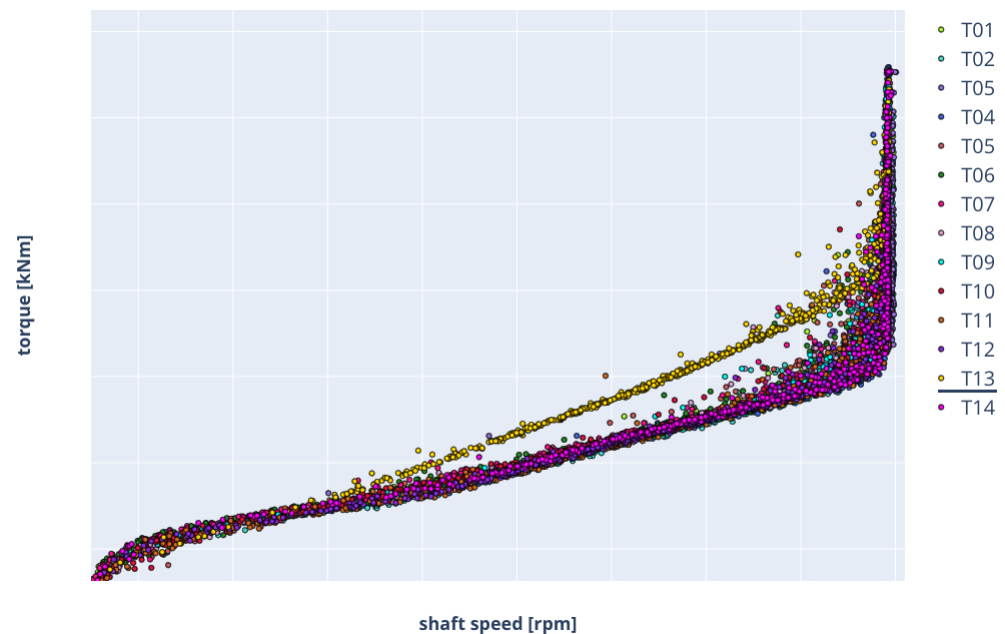


Figure 11 Torque versus shaft speed curve for region 2 operation, data from the wind farm used in this project.

The significant differences observed for turbine T13 are unlikely to be causing a loss of power, since this would have been quickly detected and corrected (and is not visible in the power curve of T13 compared to other turbines). However, it cannot be due to a malfunctioning shaft speed sensor, so deserves further investigation.

Less error-prone techniques than power curve analysis are also available³, which have often been in use for many decades. However, these often rely on wind flow modelling (not available for the wind farm under study here) to predict wind flow and turbine performance.

For this reason, the project investigated the concept of creating a 'social network' of turbines, which benchmark themselves against each other. Comparison of differences or ratios in SCADA parameters, between turbines which are close enough to be operating in similar conditions, should enable the truth to be established: detecting both trends and sudden changes in time, and attributing these changes to a likely cause.

A similar concept was only recently considered in the literature—in parallel with this project—for the purpose of collaborative wind turbine control⁴. The goal of that work was the adjust the turbines' yaw controllers to minimise yawing time and yaw misalignment. The problem was presented as an optimisation task, with each turbine iteratively determining its own local wind direction and bias, based on a comparison with local turbines.

That work is very interesting in the context of this project, and the conceptual framework for consensual analysis has some similarities, despite the differing

³ A. Albers, "Efficient Wind Farm Performance Analysis", Deutsche Wind Guard Consulting, 2004

⁴ J. Annoni et al., "Wind direction estimation using SCADA data with consensus-based optimization", Wind Energ. Sci., 4, 355–368, 2019

objectives. As stated by Annoni et al., the wind direction estimator is a potential technique which could be applied inside the framework presented here (although we demonstrate another, based on statistical inference).

Finally, regarding the use of nacelle lidar for the establishment of yaw misalignments, the theoretical basis for this is established in documents such as Vindforsk technical report 2016:298⁵.

3.2 Social Analysis Framework

The first step in establishing a social network for the turbines is to use the provided locations and hub heights (above a fixed reference such as Mean Sea Level) to establish distance and interaction matrices:

- Vertical separation (m): $z_{A,B} = z_B - z_A$
- Horizontal separation (m): $s_{A,B} = \sqrt{(y_B - y_A)^2 + (x_B - x_A)^2}$
- Wind direction for wake impingement (°): $\theta_{A,B} = \left(\frac{180}{\pi}\right) \tan^{-1} \frac{x_B - x_A}{y_B - y_A}$

These are defined such that the first turbine in the pair (A) is considered the reference. For example, a positive vertical separation indicates that turbine B is higher than turbine A.

Importantly (and assuming the locations are given in a projection which creates a locally Cartesian system, with true North aligned with the vertical axis) this information provides a source of truth for the wind direction.

When the power ratio of two turbines aggregated over time is plotted against direction, each relevant wake should be visible as a fluctuation in that ratio. The direction of greatest increase/decrease should, barring systematic deviations to the flow caused by orography, indicate the relative direction of the wind turbine, which can be compared against the expected direction from the known turbine positions.

Figure 10 illustrates this for the power ratio of a hypothetical turbine pair. A gradient descent optimisation—such as the *optimize.minimize* function from *scipy* in Python—can be used to fit the expected wake profiles and thus determine the calibration error in the nacelle-measured wind direction. This procedure already has some currency in the wind energy sector.

Unfortunately this important source of truth was unavailable here, since the highly bi-directional nature of the wind (easterly and westerly only) at the particular site used in the project meant that both sides of the wake profile could not be seen.

⁵ U. Turkyilmaz, J. Hansson and O. Undheim, "Use of remote sensing for performance optimization of wind farms", Kjeller Vindteknikk, 2016

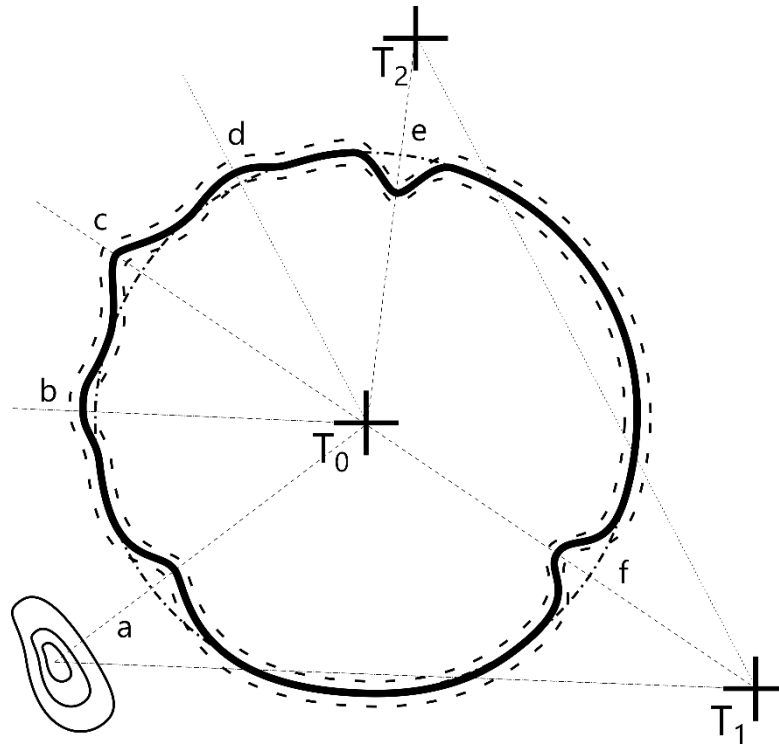


Figure 12 Schematic of the detection of true turbine angles from wake effects. The thick solid line shows the power ratio of T_0/T_1 . The dashed lines illustrate the uncertainty (standard error). The wake of T_0 on T_1 is seen at direction c, and vice versa at direction f. Another turbine T_2 wakes T_0 at location e, and T_1 at location d. Finally, upstream orography reduces the power produced by turbine T_0 at direction a, and T_1 at direction b.

The key improvement over standard SCADA analysis now developed is the recognition that the data forms a time series, such that neighbouring points are likely to have a correlation. Since a wind farm owner's SCADA system already stores the data, the concept behind the design of *this* system is to parse each set of 10-minute aggregated data as it arrives, in order to perform an update of its inferences about the states of the turbines and wind farm (in less than 10 minutes). This can be used to drive a real-time dashboard display of the wind farm, as outlined in Figure 11. The following subsections elaborate this framework.

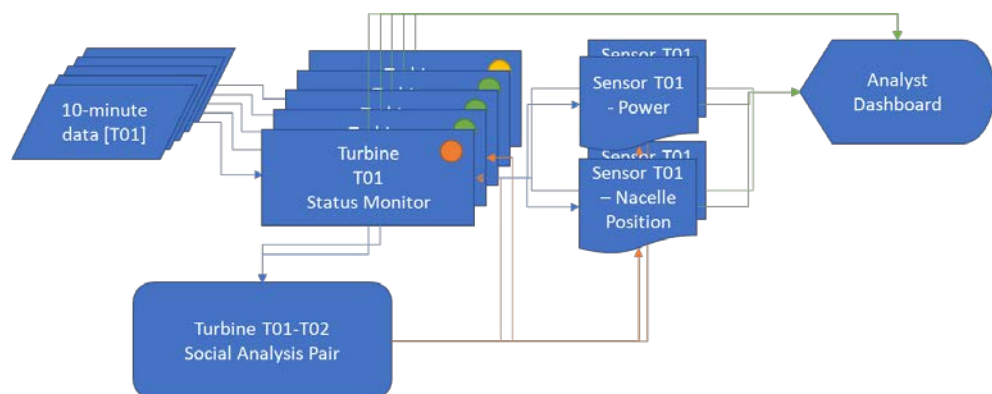


Figure 13 Structure of updating process using social statistics: blue arrows: propagation of SCADA data; orange arrows: propagation of status and calibration updates; green arrows: propagation of information to user

3.2.1 Independent turbine inference

First, a set of turbine and sensor statuses are defined, and then are set using the latest data (without yet considering other turbines). The implementation here assumes a Markov process, i.e. that a status is stored, and can be updated or retained at each time step. Previous statuses are forgotten once overwritten, thus inference of a new status can only depend the new information received and the previous status. However, at the minor cost of additional memory, trends could be remembered and used, for example to fill in missing data. The state inference process applied here is shown in Figure 12.

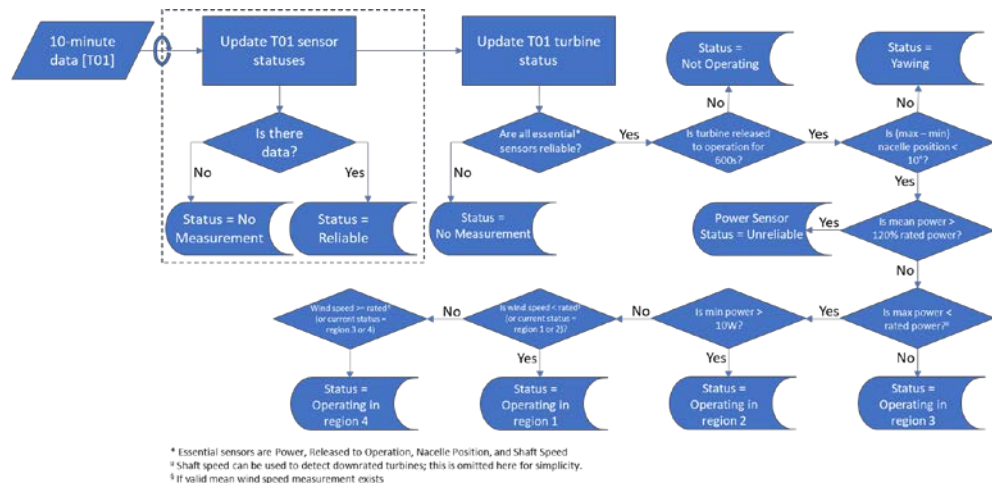


Figure 14 Turbine and sensor status analysis performed at each timestep

As can be seen, in the situation that a power reading is far higher than would be expected for the turbine, that sensor's status is overridden to be 'unreliable'. This then flags it when later processes attempt to use it (in our implementation, it is treated as if there is no measurement). Further, the turbine's status in this case is not updated, i.e. the most logical choice is to retain the status set when the latest valid measurement was received.

Sensor statuses used here are:

- 1 No measurement
- 2 Reliable
- 3 Unreliable

Turbine statuses used here are:

- 1 No measurement (i.e. SCADA is not working correctly)
- 2 Not operating
- 3 Yawing
- 4 Operating in region 1
- 5 Operating in region 2
- 6 Operating in region 3
- 7 Operating in region 4

This process chart can be improved over time by engineers and asset analysts according to their specific knowledge of the wind farm, turbines and sensors. Additional statuses can be added (for instance "downrated", or "region 2a/b/c") based on other logical rules or available SCADA tags.

3.2.2 Turbine pair inference

Once every turbine's status and sensor data are updated, this data is used to update a network of turbine pairs, which each compare the ratio or difference of a particular sensor (e.g. power, or nacelle yaw position).

The first step is to automatically create useful turbine pairs. The method chosen here is to allow the user to specify:

- the maximum number of pairs for each turbine η_{max} ;
- the maximum horizontal distance between a pair $\bar{d}_{xy,max}$; and
- the maximum vertical distance between a pair $\bar{d}_{z,max}$.

The following algorithm is then used to generate at least one pair per turbine. For each turbine it creates pairs with each turbine which meets the above criteria, until the maximum allowed number is reached, nearest horizontal distance first. If no turbines meet the criteria, it first tries to identify turbine which meets the horizontal distance criterion and violates the vertical criterion least. If this fails, it tries the opposite approach: from those turbines which are close enough vertically, it finds the one which least violates the horizontal distance. Finally, if there is no turbine matching either criteria, it simply chooses the one closest horizontally.

- Enter η_{max} , $\bar{d}_{xy,max}$, $\bar{d}_{z,max}$
- $P \leftarrow []$
- For each turbine T_A :
 - $p, d_{xy}, d_z \leftarrow [], [], []$
 - For each other turbine T_B :
 - Calculate horizontal distance between turbines $\bar{d}_{xy,AB}$ and add to d_{xy}
 - Calculate vertical difference between turbines $|\bar{d}_{z,AB}|$ and add to d_z
 - If P contains p_{BA} , add p_{BA} to p
 - Else create new turbine pair p_{AB} and add to p
 - Sort p, d_{xy}, d_z in ascending order of d_{xy}
 - $p_{final} \leftarrow []$
 - For each pair p_j in p :
 - If $d_{xy,j} \leq \bar{d}_{xy,max}$ and $d_{z,j} \leq \bar{d}_{z,max}$, add p_j to p_{final}
 - If p_{final} contains η_{max} pairs, end loop
 - If p_{final} contains 0 pairs:
 - $\bar{d}_{z,min}, i \leftarrow 10^{10}, -1$
 - For each pair p_j in p :
 - If $d_{xy,j} \leq \bar{d}_{xy,max}$ and $d_{z,j} < \bar{d}_{z,min}$:
 - $\bar{d}_{z,min} = d_{z,j}$
 - $i = j$
 - If $i \geq 0$, add p_i to p_{final} , else:
 - $\bar{d}_{xy,min}, i \leftarrow 10^{10}, -1$
 - For each pair p_j in p :
 - If $d_{z,j} \leq \bar{d}_{z,max}$ and $d_{xy,j} < \bar{d}_{xy,min}$:
 - $\bar{d}_{xy,min} = d_{xy,j}$
 - $i = j$
 - If $i \geq 0$, add p_i to p_{final} , else:
 - Add p_0 to p_{final}
 - Add p_{final} to P

The next step is to implement inference algorithms, to determine information such as:

- Whether a sensor has likely shifted calibration (suddenly or gradually), relative to a neighbour, and if so, how much.
- Which directions exhibit wake (or orography) effects, and by how much.
- Whether a turbine's sensor value is consistently offset from that of its neighbour's (for example, power loss).

Sensor re-calibration is worth considering here, as it can be inferred with the following method:

- 1 Create a regression model for the difference (or ratio, if more logical) in the sensor values of a given turbine pair with direction. This can use the reference turbine's wind vane, nacelle direction sensor, or the reference met mast; whichever combination of these is used, inference of a reference wind velocity is important for most analyses. This is best modelled with a Gaussian process (even though data storage is required), although other flexible methods which provide uncertainty on predictions should work.
- 2 When each valid update is received (perhaps filtered by the turbine's operating state as well as the sensor's state), use it to update the regression model.
- 3 Each time step, or less frequently—perhaps on demand—reset or update any inferred re-calibration values that have been assigned to sensors, based on the regression model (see section 3.4 for a specific example).
- 4 Usually, a sudden change in calibration occurs during maintenance. Therefore, detect such periods using a change in turbine status from 'out of operation' to 'in operation'. When this occurs, archive the old regression model and start creating a new temporary one.
- 5 As each new valid data point is collected, update the temporary model, and run a t-test to check whether the mean μ of the new model is likely to be significantly different to the old (locally to the direction just acquired):

$$p = 1 - T \left(\frac{\mu_{old} - \mu_{new}}{\sqrt{\sigma_{new}^2/n_{new} + \sigma_{old}^2/n_{old}}}, n_{new} + n_{old} - 2 \right)$$

where T is the cumulative distribution function of the Student's t distribution, σ is the standard deviation of the predicted mean, and n is the number of measurements used to make the prediction. It may also be desirable to enforce a minimum change $|\mu_{old} - \mu_{new}|$, or a maximum number of values n_{new} to collect before making the decision, since p values can often become significant (0.05 is often used as the threshold) once enough data is collected, even for very small changes.

- 6 If a significant change is detected, abandon the old model and treat the temporary model as the new one. If no significant difference is found, add the data from the temporary model to the old model.

3.2.3 Wind farm inference

Finally, the information gathered by the turbine pairs can be compared across the wind farm to obtain more global information. This should include:

- 1 Assigning global data (such as from meteorological masts);
- 2 Attributing sudden or gradual changes measured by pairs back to individual turbines, by finding the most likely explanation looking at the whole network;
- 3 Assigning sensor calibration values by finding the optimal choices across the network;
- 4 Determining individual turbine's likelihood of exhibiting a particular degradation mechanism.

In section 3.4, an example of the 2nd and 3rd inference algorithm is developed, focusing on yaw direction misalignment. It is important to build up a library of explanatory models for different failure modes, since they are likely to exhibit in similar ways but with subtly effects on different combinations of sensors' data.

3.3 Software Structure

The usefulness of a SCADA analysis framework lies not only in its theoretical power on paper, but also in its simplicity and robustness to be implemented in software. Thus, effort was put in this project into designing and implementing a software architecture which could be copied by wind farm analysis organisations. This architecture enables individuals to try particular inference algorithms without disrupting (or needing to understand) the overall software.

The social network was thus implemented in Python using an Object Oriented approach, in order to simulate providing each time stamp of data and then running the inference methods described in the previous section (on the general framework), and the following section (on yaw direction specifically). As well as a robust architecture, it was important to test whether the inference algorithms were indeed fast, such that they can be easily run within the 10-minute gap between updated SCADA signals.

The Observer software engineering pattern was used as the basis of the design. In this pattern, which is commonly used to architect event-driven software, messages are passed from each higher-level object (here called a Sender) to lower-level objects which is registered to it (here called Receivers).

Receivers have one method by default:

- 1 *update(data)*: this is an abstract method, meaning that classes implementing the Receiver must specify what actions should be taken when receiving data.

Senders have by default three methods:

- 1 *register_receiver(receiver)*: this adds a new Receiver to the internally maintained list of objects which should receive messages;
- 2 *unregister_receiver(receiver)*: this removes a Receiver from the list, avoiding memory leaks;
- 3 *notify_receivers(data)*: This forwards the data to all registered Receivers by calling their *update(data)* method.

These concepts are implemented as follows for the wind farm SCADA analysis:

- a *Sensor* is a Receiver. These are created first and then registered to:
- a *Turbine*, which is both a Sender and Receiver. These are created and then registered to:
- a *WindFarm*, which is a Sender.

Each 10-minute set of SCADA data is sent, looping over each turbine, into the *WindFarm*'s *notify_receivers* method. This automatically forwards it to the *Turbines*, which decide whether or not it is appropriate for them, and if so first use their *notify_receivers* method to forward it to all *Sensors*.

The Sensor's *update* method immediately determines its status, as well as updating its current data, if deemed reliable. Then the Turbine's *update* method continues by determining its own status (all as described in section 3.2.1).

The Turbine and Sensor classes each have a supporting Properties class, which contains static information, such as location and power curve for a Turbine, and variable type for a Sensor. While inferred information, such as a calibration offset, could be stored here, it was determined that it is preferential to create a VirtualSensor class, which extends the Sensor class, allowing decoupled storage and application of a calibration offset to the data already in the relevant Sensor class.

A TurbinePairManager automatically creates a network of TurbinePair objects for any desired sensor variable, and are stored in the WindFarm. These are manually updated once the individual turbine's updating is completed.

The full Unified Modelling Language (UML) class diagram is shown in Figure 13. This structure is entirely generic; the logic for detecting and assigning recalibration values is written inside methods such as *TurbinePairManager.recalibrate_all*. It is recommended to make the TurbinePairManager class into a Builder pattern, such that it constructs and returns a specific type of class with logic that depends on the particular sensor type and failure mode being analysed.

Finally, by implementing a logging system, any detected change in calibration (which caused a TurbinePair to discard its previous model and start afresh, as described in section 3.2.2) is automatically logged to a text file for further analysis.

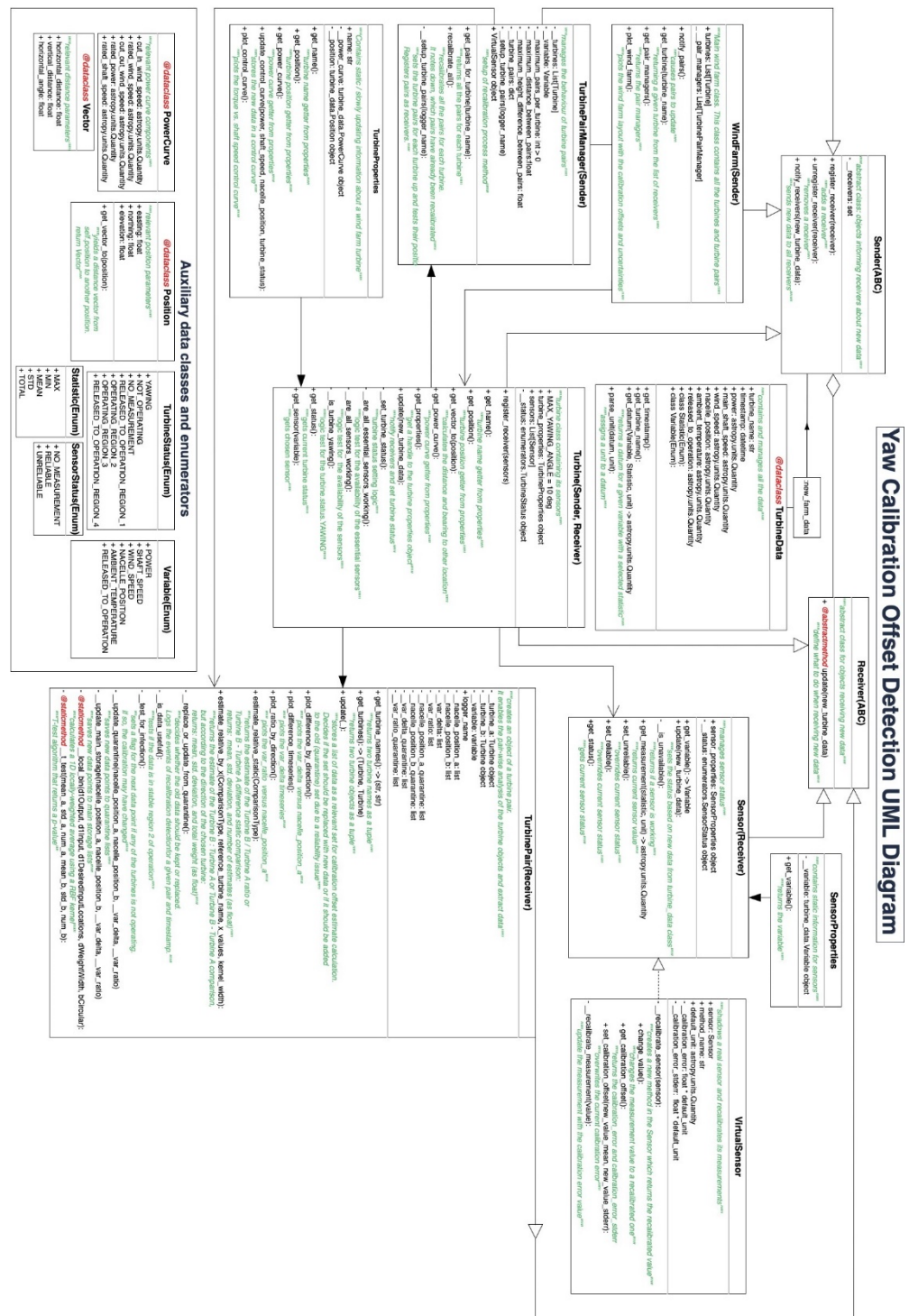


Figure 15 Unified Modelling Language (UML) class diagram for the Python code created for this project

3.4 Yaw Analysis

In this project, because of the existence of a Lidar campaign which directly measured yaw misalignment, the decision was taken to focus on this cause of turbine power loss. Thus, best efforts are now made to develop a model able to infer this.

The mean measured nacelle yaw angle of a turbine θ over a 10-minute period, when in power production in region 2, can be assumed to be determined by the following combination of effects, illustrated in Figure 14:

$$\theta = \theta_{local} + \delta\theta_m + \delta\theta_c + \varepsilon$$

In words, the turbine angle from True North θ should be aligned with the direction from True North from which the wind is coming θ_{local} . However, this is affected by:

- $\delta\theta_m$ —a systematic error in the control system which causes the turbine to be aligned at an angle to the wind;
- $\delta\theta_c$ —a pure measurement error, whereby the North reference of the sensor is not equal to True North;
- ε —a random error (assumed here to be Normally-distributed and zero-mean) caused mostly by the delay in the control system to adjust to the turbulent fluctuations in wind direction experienced during the 10-minute period.

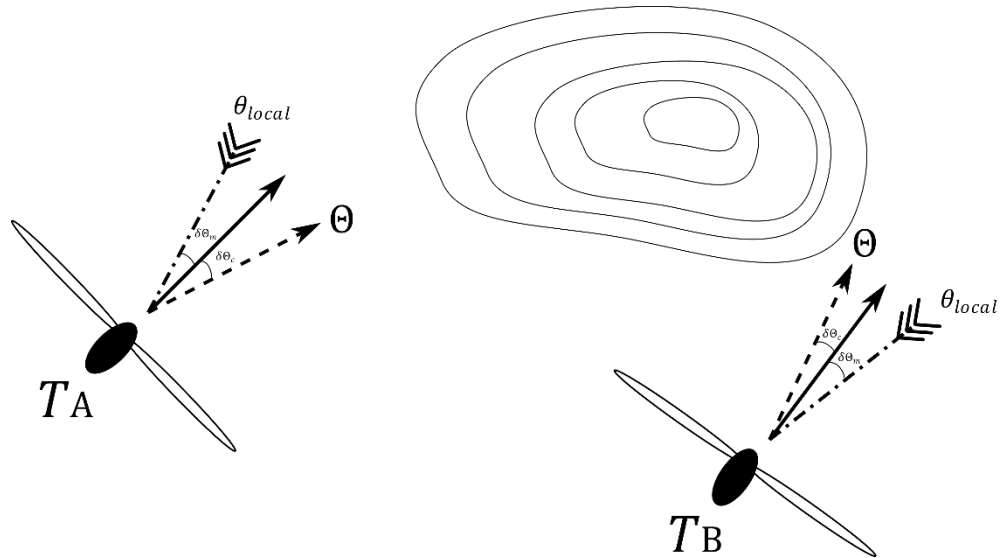


Figure 16 Illustration of nacelle yaw position measurement effects

The reason it is specified here that these assumptions only hold true in region 2 (partial load) operation, is because in region 3 (full power) it may be that the control system is programmed to yaw less, in order to reduce wear on the yaw bearing. It is also possible that this is not the case, however, and then region 3 may prove to provide more stable data for yaw direction comparison. In any case, 10-minute periods with significant dynamic yawing towards a new direction are excluded, as consistent operation is desired for reliable statistical inference.

By subtracting the measured yaw position of two turbines, we end up with the following equation:

$$\Delta\theta_{AB} = \theta_{local,B} - \theta_{local,A} + \delta\theta_{m,B} - \delta\theta_{m,A} + \delta\theta_{c,B} - \delta\theta_{c,A} + \varepsilon$$

For a pair created according to the methodology in section 3.2.2, such that the turbines are quite close together and at similar elevations, it is reasonable to assume that the relative local wind directions may be systematically affected by orography, but that such a variation—which is related to inflow direction—can be averaged over all directions to have zero mean:

$$\int_0^{360} (\theta_{local,B} - \theta_{local,A}) d\theta \sim \mathcal{N}(0, s_\theta)$$

Where no data are available for a substantial portion of the compass (as in this wind farm, where the wind flows bidirectionally), this assumption may be invalidated and result in an unknown bias (which can be treated as uncertainty) on the results.

Calibration errors are assumed piecewise constant (i.e. no variation by direction) in-between step changes in time (which can occur after maintenance, as previously described). This assumption may be refuted if hysteresis effects related to large yaw changes are proven; however, there is currently insufficient corroboration for this idea.

Finally, yaw misalignment with the incoming wind is also generally assumed constant, although unlike a calibration error it should result in a loss of power. It is commonly stated that the angle acts on the power available to the turbine as follows:

$$P(\delta\theta_m) = \frac{1}{2} \rho A (v \cos \delta\theta_m)^3$$

Thus a power loss of $\cos^3 \delta\theta_m$ is assumed. However, this is overly conservative, and a loss of less than $\cos^{1.5} \delta\theta_m$ has been observed by wind turbine manufacturers in practice. Not knowing the true value of the exponent here, and particularly having little information on how it varies with air density, wind shear (which is not measured), and so forth, means that power output differences cannot meaningfully be used to infer a particular yaw misalignment angle. This is particularly true because any measured power losses could come from any number of degradation modes to the turbine, for instance extra roughness on the blades, or wear in the gearbox (although these degradation modes could exhibit a varying power ratio with wind speed).

Thus we currently have two inextricable effects—miscalibration and misalignment—which can separately or in combination cause a disagreement in nacelle position when averaged over all directions.

We now turn to multiple turbine pair data for a wind farm, as per section 3.2.3, to search for a solution which we can apply. Figure 15 illustrates a theoretical example, where the true (calibration + misalignment) offset is marked on the map. We will measure the directionally-averaged pairwise differences shown in Table 3.

Table 3 Illustration of mean pairwise nacelle yaw position differences for the example in Figure 15.

Pair	Average Yaw Difference $\Delta\theta$ (°)
$\Delta\theta_{12}$	-15
$\Delta\theta_{13}$	-10
$\Delta\theta_{23}$	5
$\Delta\theta_{24}$	18

In order to estimate the original $\delta_i = \delta\theta_{m,i} + \delta\theta_{c,i}$ values, which are provided in Figure 15, we will apply Bayesian updating.

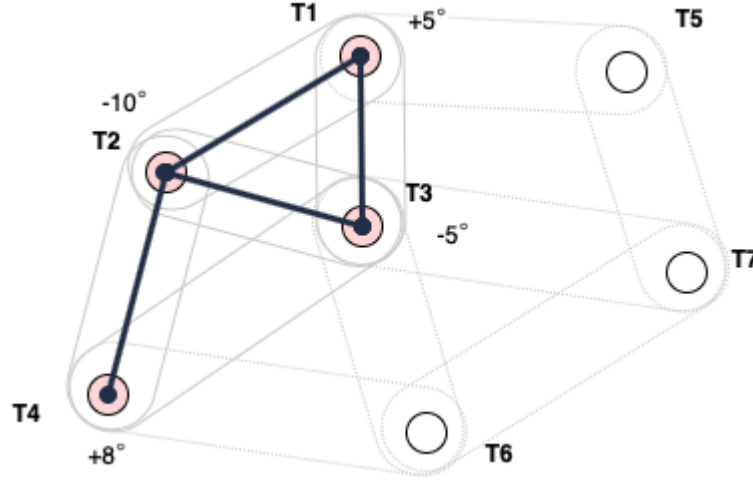


Figure 17 Illustration of a social turbine network for Bayesian inference of true nacelle position

Bayes' Rule states that the posterior (updated) probability $p(\mu|D)$ of the mean μ taking a certain value, given some data D , is proportional to the prior (previously assumed) probability of that value $\pi(\mu)$, multiplied by the likelihood of the data given that the value is correct $L(D|\mu)$. In this case we are observing correlated values from the pairs, thus the updates need to be simultaneous:

$$p(\mu_A, \mu_B | \Delta\theta_{AB}) \propto \pi(\mu_A)\pi(\mu_B)L(\Delta\theta_{AB} | \mu_A, \mu_B)$$

First, a prior $\pi(\mu)$ is needed to encode our assumption that—before seeing any data—the yaw miscalibration lies somewhere in the range $[-180^\circ, +180^\circ]$. This could be a uniform distribution, however, using a Normal distribution results in the approximate inference method we are going to use working better. Thus we need to determine an equivalent variance for a zero-mean Normal distribution.

The variance of a uniform distribution supported between a and b is $\frac{1}{12}(b - a)^2$. This gives $\sigma_0 \approx 104$ for our range of 360° . For a zero-mean Normal distribution with this variance, the cumulative probability of a value less than -180° would be $\Phi\left(-\frac{180}{104}\right) \approx 0.042$. Thus 8% of the probability would lie outside the region of acceptable values.

This being unacceptable, instead we determine the variance σ_0^2 which gives less than 0.1% probability density outside $[-180^\circ, +180^\circ]$:

$$1 - 2 \Phi\left(-\frac{180}{\sigma_0}\right) < 10^{-4}$$

$$\therefore \sigma_0 \lesssim 46$$

For safety, $\sigma_0 = 40$ is used here.

The inference is performed using Markov Chain Monte Carlo (MCMC), a standard approach for performing inference in Bayesian statistics. The Python *pymc3* package is used, where the following code snippet performs the required sampling to calculate the results in Table 4 from the inputs in Table 3.

```

import pymc3 as pm
with pm.Model() as model:
    t1 = pm.Normal('t1', mu=0, sigma=40.)
    t2 = pm.Normal('t2', mu=0, sigma=40.)
    t3 = pm.Normal('t3', mu=0, sigma=40.)
    t4 = pm.Normal('t4', mu=0, sigma=40.)
    diff12 = pm.Deterministic('t2-t1', t2 - t1)
    diff13 = pm.Deterministic('t3-t1', t3 - t1)
    diff23 = pm.Deterministic('t3-t2', t3 - t2)
    diff24 = pm.Deterministic('t4-t2', t4 - t2)
    obs12 = pm.Normal('obs12', mu = diff12, sigma=0.5, observed=-15)
    obs13 = pm.Normal('obs13', mu = diff13, sigma=0.5, observed=-10)
    obs23 = pm.Normal('obs23', mu = diff23, sigma=0.5, observed=5)
    obs24 = pm.Normal('obs24', mu = diff24, sigma=0.5, observed=18)
    trace = pm.sample(1500, init="adapt_diag")
print(pm.summary(trace))

```

The results are shown in Table 4. As can be seen, the final estimates are within 0.2° of the true values. Note that the posterior uncertainty is high, even though in this case the input data is specified as having only 0.5° uncertainty.

Table 4 Calculation of turbine yaw offsets using Markov Chain Monte Carlo, see Figure 15.

Turbine	True Yaw Offset δ_i (°)	Estimated δ_i (°)	Error (°)
1	+5	4.8 ± 19.3	-0.2
2	-10	-10.2 ± 19.3	-0.2
3	-5	-5.1 ± 19.3	-0.1
4	+8	7.8 ± 19.3	-0.2

Inferring calibration error in this way is valuable for the industry, as it quickly highlights which turbines could benefit from re-calibration, which then enables further analysis.

However, it is clearly not possible to separate out calibration error from true yaw misalignment with this method. If all turbines have their yaw position sensors corrected to True North (either by inferring this error from the nacelle wind vane sensor or, preferably, by conducting maintenance to align them) then the pairwise average yaw misalignment reduces to the following:

$$\overline{\Delta\theta_{AB}} = \delta\theta_{m,B} - \delta\theta_{m,A}$$

A single source of truth now suffices to propagate yaw misalignment estimates across the whole network. This can come from a nacelle-mounted Lidar installation, which can be on any turbine, although a turbine with a large probability of being unwaked and which is paired with several others should be prioritised.

4 Results and Analysis

4.1 Data preparation

As well as the turbine SCADA data, Fortum supplied data from several campaigns of a ZephIR (now ZX Lidars) nacelle-mounted lidar. All data used for this analysis are 10-minute averaged statistics.

The SCADA data came in the form of multiple Microsoft Access databases. The lidar data is converted from the binary ZPH files to CSV using Waltz v4.7. Thereafter both data sets are imported and time-synchronised into a Python *pandas* DataFrame.

4.2 Lidar data analysis

Yaw misalignment measurements were performed by Fortum using a nacelle lidar in four campaigns covering three turbines:

- 1 T11: 2018-03-15 15:50 - 2018-04-15 23:50
- 2 T13: 2018-04-17 09:40 - 2018-05-13 23:50 (campaign a)
- 3 T05: 2018-05-24 10:00 - 2018-07-16 23:50
- 4 T13: 2018-07-27 11:30 - 2018-10-10 23:50 (campaign b)

This data is now analysed to determine the yaw calibration error and then the yaw misalignment for each turbine. The processing methodology described by ZX Lidars is reimplemented as closely as possible.

4.2.1 Calibration error

To obtain an estimate of the calibration error, we use the lidar's internal compass and compare it to the turbine's nacelle position obtained from the SCADA data. However, while a constant error was expected, Figure 16 reveals a strong directional dependency (which is repeated on other turbines). A simple sinusoidal fit of the data estimates an average offset of -0.2° , which fits well with the required assumption that the nacelle yaw direction sensor calibration error is zero.

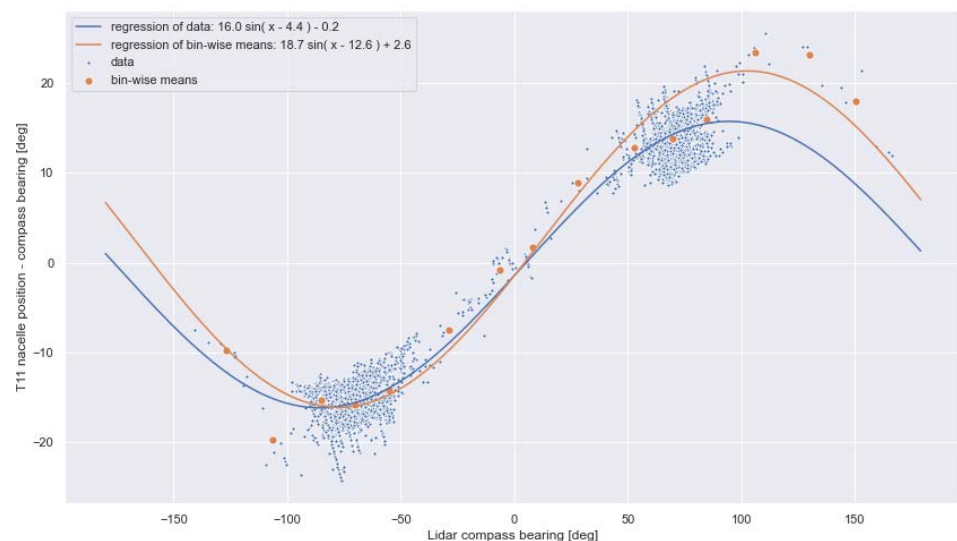


Figure 18 Disagreement between the lidar compass and the nacelle direction of turbine 11.

4.2.2 Yaw misalignment

The 10-minute average yaw misalignment is computed by the nacelle lidar. We use the results for the “fit-derived” algorithm, which uses all measurement points along the scanned circle, as visualised in Figure 17. The wind speed is measured about half a rotor diameter upstream.

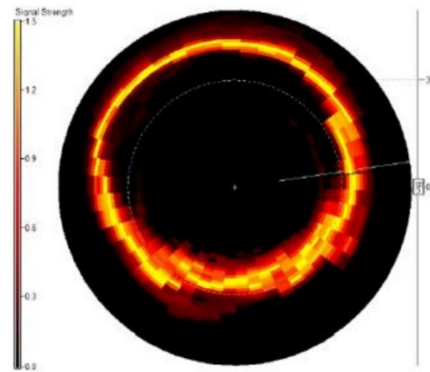


Figure 19 Visualisation of the fit-derived algorithm used by the ZX lidar to determine the yaw misalignment.

The resulting yaw misalignment fluctuates, because it is, in part, caused by the inherent laziness of the yaw controller. We are interested in the systematic offset that remains after averaging out this normal fluctuation.

For each lidar campaign we compute the average yaw misalignment measured by the nacelle lidar. Prior to averaging the lidar data is filtered using the following criteria:

- Turbine must be operational throughout the entire 10-minute interval.
- The power produced must be at least 5kW.
- The nacelle direction must be inside the measurement sector: 60° - 120°. This is the free wake sector for turbines 11 and 13, but for consistency it is also applied to T05.

The resulting (cumulative) average yaw misalignment for each campaign is shown in Figure 18, Figure 20, Figure 22 and Figure 23. The blue graph traces the cumulative average yaw misalignment, while the green traces the standard error. At times when the measured data does not satisfy the filter criteria outlined above – most notably when the wind directions are not inside the measurement sector – the averages are not updated, resulting in a horizontal section of the graph. The final average values obtained for each campaign listed in Table 5.

For the T11 and T13a campaigns, Fortum shared yaw misalignment analysis results obtained under more elaborate filter conditions. These results are shown in Figure 19 and Figure 21. These yaw misalignment values, also shown in Table 5, are very close to those obtained in our analysis.

Knowing the yaw misalignment on a single turbine, we will use social statistics to extrapolate that misalignment to all turbines in the farm. Because we have measured the misalignment on more than one turbine, we can verify those extrapolations.

Table 5 Yaw misalignment for each lidar campaign.

Lidar campaign	Yaw misalignment		Standard error	
	TNO	ZX Lidars	TNO	ZX Lidars
T11	-4.35°	-3.99°	0.11°	0.11°
T13 a	-6.80°	-6.68°	0.06°	0.06°
T05	-5.68°	-	0.24°	-
T13 b	-5.20°	-	0.07°	-



Figure 20 Cumulative average yaw misalignment for turbine 11.

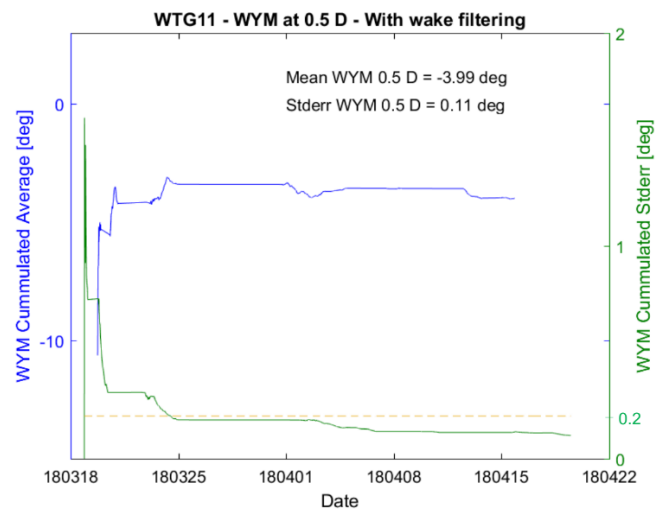


Figure 21 Cumulative average yaw misalignment for turbine 11 as calculated by ZX Lidars.



Figure 22 Cumulative average yaw misalignment for turbine 13 (campaign a).

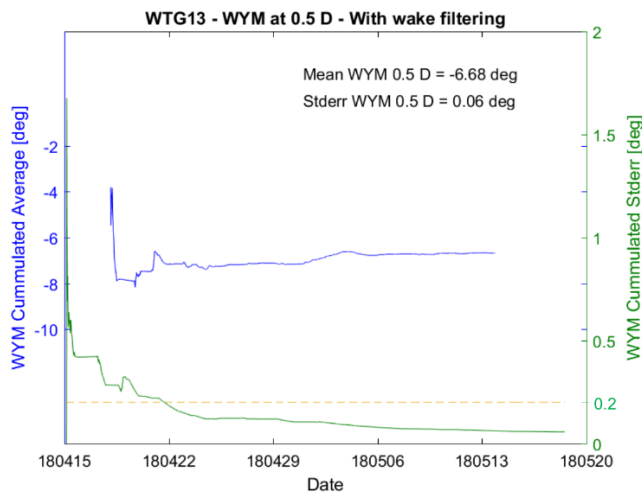


Figure 23 Cumulative average yaw misalignment for turbine 13 (campaign a) as calculated by ZX.



Figure 24 Cumulative average yaw misalignment for turbine 5.

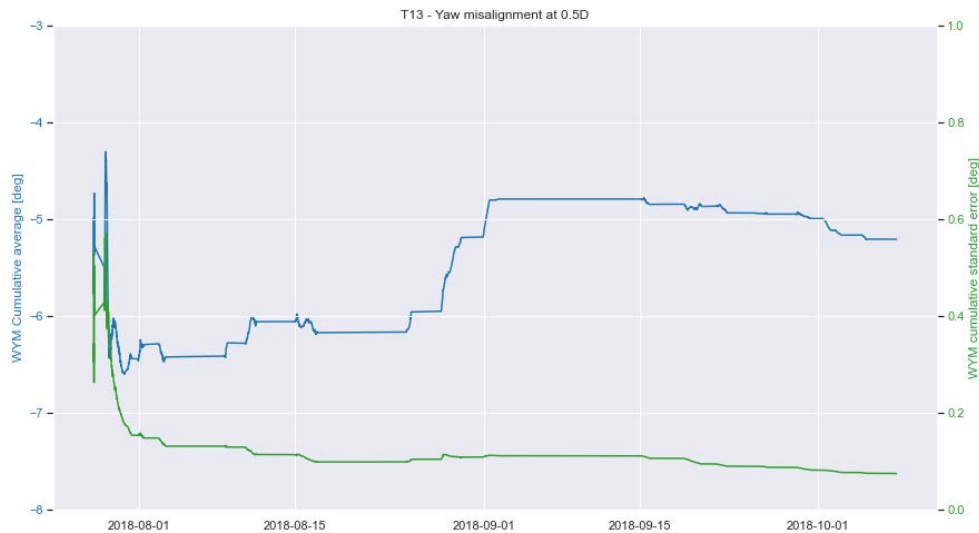


Figure 25 Cumulative average yaw misalignment for turbine 13 (campaign b).

As can be seen from Figure 18 - Figure 23, the estimate of yaw misalignment consistently changes whenever new data is received to update it. This change is greater than the standard error derived from the cumulative number of data points acquired, suggesting that the measurement is not stationary. This is a concern which should be addressed in other projects, to determine what may cause such instability.

4.3 Applying social statistics

The turbine pair manager is configured to create pairs of turbines not further than 750m apart and not differing more than 10m in elevation. This resulted in the pairing of turbines shown in the first two columns of Table 6.

The simulation is run using SCADA data starting on 2018-01-01. The estimated yaw errors for all turbine pairs are logged at the end date of each of the lidar campaigns. Figure 24 shows a typical result for the difference of the nacelle positions of a turbine pair, as a function of the nacelle position of one of the turbines in the pair. We can see some variation, which is to be expected, because the yaw controller of each turbine operates independently of the other turbines. Again, we are interested in the systematic deviation. It can also be seen that the average difference is different for the easterly and westerly wind direction. This may well be caused by differences in inflow conditions, such as wakes and orography.

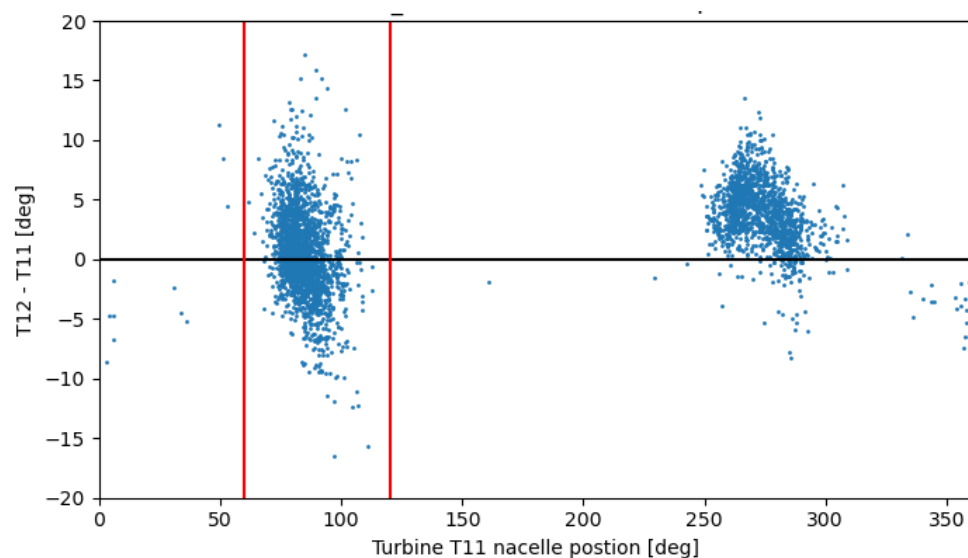


Figure 26 Example of yaw error estimation. The nacelle position difference is compared for turbine pair T11 and T12. The easterly measurement sector is indicated by red bars.

The average yaw differences for each pair are computed at the end of each lidar campaign. This is done separately for the easterly ($60^\circ - 120^\circ$) and westerly ($240^\circ - 310^\circ$) wind sectors. The results are reported in Table 6 (easterly sector) and Table 7 (westerly sector).

There are two main conclusions to be drawn straight away:

- 1 Most of the turbine exhibit systematic differences between easterly and westerly wind directions. Given the layout of the wind farm with turbines aligned closely to the wind direction and therefore likely experiencing wakes in one of these directions, this is not surprising.
- 2 De-calibration (or intentional re-calibration) events can be seen to have a sudden effect on some of the turbine pair differences during the 6-month period. However, some of the pairs have very stable values. Given the stability of all turbine T07 pairs except for its relationship with T11, this implies that a re-calibration event occurred to T11 between the T13(a) and T05 Lidar campaigns.

Table 6 Yaw differences between turbine pair A and B for all four lidar campaigns, easterly wind direction sector (60° – 120°). Bold entries are used for inference, since they are less waked.

Turbine		Mean difference				Standard error			
A	B	T11	T13(a)	T05	T13(b)	T11	T13(a)	T05	T13(b)
11	12	0.50	0.30	0.36	0.28	0.08	0.06	0.05	0.04
11	10	-4.97	-4.16	-4.22	-4.02	0.49	0.12	0.08	0.04
11	7	2.24	2.15	0.70	1.25	0.09	0.06	0.25	0.06
12	13	-0.16	-0.02	0.01	-0.06	0.05	0.04	0.04	0.03
12	10	-3.10	-3.96	-4.14	-2.28	0.26	0.11	0.10	0.16
7	6	0.52	0.63	0.68	0.79	0.05	0.04	0.04	0.04
7	8	-1.31	-1.70	-1.80	-2.17	0.27	0.13	0.11	0.10
7	3	-0.43	-0.25	-0.24	-0.17	0.06	0.05	0.04	0.04
13	14	1.10	-0.31	-0.23	-0.33	0.14	0.07	0.06	0.10
6	5	-4.53	-4.55	-4.57	-4.65	0.04	0.04	0.04	0.03
8	9	8.48	8.99	8.92	9.21	0.15	0.08	0.36	0.07
5	4	7.67	7.53	7.57	7.41	0.04	0.04	0.03	0.03

Table 7 Yaw differences between turbine pair A and B for all four lidar campaigns, westerly wind direction sector (240° – 310°). Bold entries are used for inference, since they are less waked.

Turbine		Mean difference				Standard error			
A	B	T11	T13(a)	T05	T13(b)	T11	T13(a)	T05	T13(b)
11	12	3.92	3.68	2.73	2.55	0.07	0.06	0.05	0.04
11	10	-1.25	-0.91	-1.42	-1.46	0.11	0.52	0.05	0.04
11	7	1.83	1.58	-0.45	0.92	0.08	0.08	0.18	0.07
12	13	-0.02	-0.13	-0.58	-0.65	0.06	0.05	0.05	0.04
12	10	-5.97	-5.72	-4.45	-5.12	0.10	0.09	0.06	0.25
7	6	-0.57	-0.62	-1.25	-1.29	0.07	0.07	0.05	0.04
7	8	-0.52	0.63	0.96	0.22	0.16	0.28	0.08	0.10
7	3	-0.23	-0.16	-0.17	-0.19	0.06	0.06	0.04	0.04
13	14	-0.86	-1.00	-1.07	-0.36	0.08	0.07	0.05	0.11
6	5	-6.09	-6.13	-5.73	-5.49	0.09	0.08	0.06	0.05
8	9	6.10	5.63	2.65	4.73	0.11	0.12	0.20	0.06
5	4	5.87	5.99	5.02	4.66	0.07	0.07	0.05	0.04

The MCMC Bayesian inference approach described in section 3.4 is now applied to recover predictions of the yaw error for each turbine where a pair has enough data (i.e. not turbines 1 and 2, due to frequent detected changes in calibration). In Table 8, the inference is first performed for all turbines after the T11 Lidar campaign, but without including the Lidar yaw misalignment measurement. Then the Lidar-measured value is included.

Consistent results are obtained between the two estimates; including the Lidar information shifts all the estimates by approximately +3.4°. As a result, T13 is predicted to have a yaw misalignment of -4.0° (compared with -6.8° and then -5.2° measured in its later Lidar campaigns), and T05 is predicted to have a yaw misalignment of -7.2°, compared with the -5.7° measured in the Lidar campaign on that turbine.

Table 8 Bayesian inference of turbine yaw misalignments (assuming zero calibration error) after the Lidar campaign on T11.

Turbine	Without Lidar information	With Lidar information
3	$-5.7 \pm 9.1^\circ$	$-2.3 \pm 0.2^\circ$
4	$-2.9 \pm 9.1^\circ$	$0.5 \pm 0.2^\circ$
5	$-10.6 \pm 9.1^\circ$	$-7.2 \pm 0.2^\circ$
6	$-6.1 \pm 9.1^\circ$	$-2.7 \pm 0.2^\circ$
7	$-5.5 \pm 9.1^\circ$	$-2.1 \pm 0.1^\circ$
8	$-6.0 \pm 9.1^\circ$	$-2.6 \pm 0.2^\circ$
9	$0.1 \pm 9.1^\circ$	$3.5 \pm 0.2^\circ$
10	$-10.9 \pm 9.1^\circ$	$-7.5 \pm 0.3^\circ$
11	$-7.8 \pm 9.1^\circ$	$-4.3 \pm 0.1^\circ$
12	$-7.3 \pm 9.1^\circ$	$-3.9 \pm 0.1^\circ$
13	$-7.4 \pm 9.1^\circ$	$-4.0 \pm 0.1^\circ$
14	$-6.3 \pm 9.1^\circ$	$-2.9 \pm 0.2^\circ$

An example dashboard containing the results using the Lidar information is shown in Figure 25. In this instance, the dashboard would highlight turbines 5 and 10 as being the most concerning in either case.

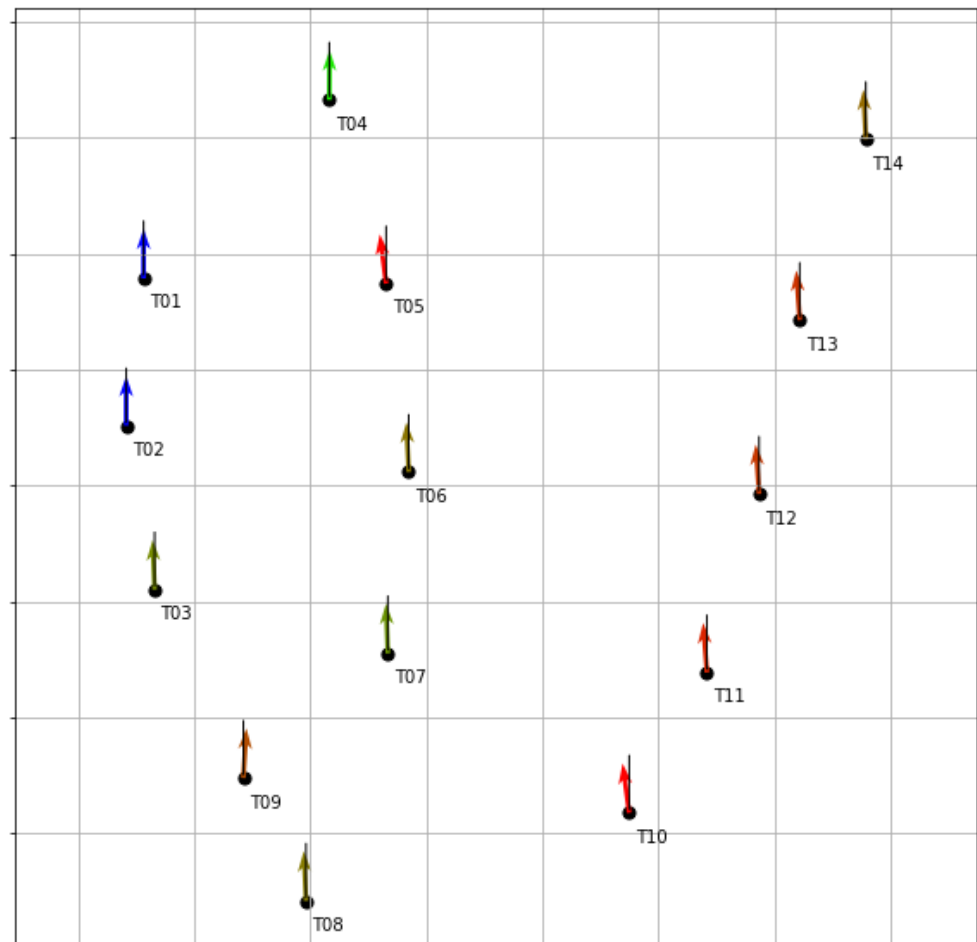


Figure 27 Dashboard showing results of turbine yaw error inferred from turbine pair results and Lidar measurement on T11 (arrow colour scales from 0° (green) to $\pm 5^\circ$ (red); blue is missing).

For comparison, the results of using the data from all three turbines after the Lidar campaign on T05 is shown in Table 9. The previous two Lidar campaign results are included, but given uncertainties of 1.5° , to account for the fact that the turbines may have been re-calibrated.

A few of the turbines' results, particularly T09, are consistent with the previous values. T05 and T10 are still of concern with the highest (negative) yaw offset. However, most of the turbines' estimates have changed by 2° or more. This is explained by the large shifts in pair values for T11-T07, T12-T10, T07-T08, T13-T14 and T08-T09.

Table 9 Bayesian inference of turbine yaw misalignments (assuming zero calibration error) after the Lidar campaign on T05.

Turbine	With Lidar information
3	$-0.0 \pm 0.3^\circ$
4	$1.9 \pm 0.2^\circ$
5	$-5.7 \pm 0.2^\circ$
6	$-1.1 \pm 0.2^\circ$
7	$0.2 \pm 0.3^\circ$
8	$1.1 \pm 0.3^\circ$
9	$3.8 \pm 0.3^\circ$
10	$-4.6 \pm 0.4^\circ$
11	$-0.5 \pm 0.4^\circ$
12	$-0.2 \pm 0.4^\circ$
13	$-0.2 \pm 0.4^\circ$
14	$-0.5 \pm 0.4^\circ$

As a final note, the speed of the MCMC inference is greatly increased with the inclusion of sources of truth for individual turbines, rather than just providing differences. Choosing one of the turbines as a reference, with zero offset, may prove a practically useful approach.

5 Conclusions and Recommendations

This project has developed and implemented an alternative method of online analysis of wind farm performance to that typically used in the industry. This conceptual framework has been called “social statistics”, as it is based on turbines comparing their behaviour to each other, enabling inference of their slowly-varying performance characteristics, freed of the many uncertainties that arise from unmeasured wind conditions.

This framework has been elaborated in this report with algorithms for creating and updating these turbine pairs and then using the whole wind farm to infer individual turbines’ behaviour. Further, a robust software architecture has been designed and tested, enabling the required event-driven processing of the 10-minute average SCADA.

The original goal of this project was to detect wind turbine underperformance automatically. Due to the particular wind farm data set provided, which included a rather valuable measurement campaign of nacelle Lidar data, the decision was taken early on to focus exclusively on implementing an algorithm for yaw misalignment detection.

However, one of the key learnings from this project is that nacelle yaw sensors can be highly unreliable. As a result, it is proven here that separation of yaw misalignment from nacelle yaw direction miscalibration is impossible without any source of independent truth to robustly re-calibrate the true nacelle direction.

Instead, an automated method for using measured differences in yaw direction between turbine pairs to infer the sum of yaw misalignment and miscalibration has been developed using Bayesian inference, and has been shown theoretically to be able to estimate correctly each individual turbine’s yaw error within a fraction of a degree. This is industrially useful as it can identify turbines for maintenance, and/or be used automatically to remove the error.

It is likely that the methods outlined here will be more successful on wind farms where the effect of orography is either known, or less pronounced. Sites with a higher spread of incoming wind directions should also be more fruitful. However, the wind farm does not need to have many turbines for social statistics to be effective, so long as the turbines are not too far apart both horizontally and vertically.

The Lidar measurements provided at the wind farm were used to reconstruct the estimated yaw misalignment. A few causes for concern were found:

- 1 The lidar compass bearing exhibited a highly directionally-dependent relationship with all the turbines’ nacelle yaw measurements, with an amplitude of approximately 30°. This may have diagnosed an issue with the yaw sensor, but could also be a concern with the Lidar’s reference.
- 2 The estimate of yaw misalignment derived from the Lidar’s measurements was dependent (to around 2°) on the choice of filters applied to the data.
- 3 The estimate of yaw misalignment did not appear to converge with time, instead consistently trending by more than the calculated uncertainty at each period with valid data.

The yaw misalignment (including miscalibration) inference was applied to the turbine pair information available at the end of the first Lidar campaign. The only difference between including the Lidar yaw misalignment estimate, and not using it, is removal of a 3.4° negative bias from the results. Thus in absolute terms the Lidar information was valuable, but the same two turbines (T05 and T10) would have registered as concerns even without any Lidar campaign.

Using those results to predict the yaw misalignments measured with Lidar campaigns on two other turbines resulted in values which were within $[-1.5, +2.8]^\circ$ of the measured values. Given the concerns about the Lidar results expressed earlier (and the length of time between the prediction and the measured result) this seems reasonable.

At the end of the third Lidar campaign, three months later, the process has been repeated. While a small number of the turbines have very similar estimates, large shifts in some of the pair differences result in changes of around 2° to many of the estimates of individual turbines' yaw errors. One potential attribution for this could be the regular changes in yaw direction sensor calibration. Another could be deliberate maintenance occasioned by the attention at the site on yaw misalignment (several of the turbines appear to be improved, including the two on which the first two Lidar campaigns were performed). However, without access to maintenance logs, the ultimate cause cannot be known with certainty.

Future extensions to this method should place more emphasis on detection of other degradation modes, such as blade damage. This can be achieved by looking more deeply at combining the power ratios of the turbine pairs with the ratios and differences of other sensors.