Wind power forecasting using random forests

by Christoffer Ahrling

LUND
UNIVERSITY

Thesis for the degree of Master of Science in Engineering
Thesis advisors: Marcus Thern, Axel Johansson

To be presented, with the permission of the Faculty of Engineering of Lund University, for public criticism at the Department of Energy Sciences on Thursday, the 12th of October 2023 at 13:15.

This degree project for the degree of Master of Science in Engineering has been conducted at the Division of Thermal Power Engineering, Department of Energy Sciences, Faculty of Engineering, Lund University.

Supervisor at the Division of Thermal Power Engineering was Marcus Thern, and assisting supervisor was Axel Johansson.

Examiner at Lund University was Jens Klingmann.

The project was carried out in cooperation with the operations department at Energy Opticon AB in Lund. Work at Energy Opticon was carried out under the supervision of Ali Moallemi.

Department of Energy Sciences
Faculty of Engineering
Lund University

ISSN: <0282-1990>
ISRN: <LUTMDN/TMHP-23/5552-SE>

Typeset in LaTeX
Lund 2023

Contents

List of Figures

List of Tables

Sammanfattning

I studien undersöktes hur maskininlärningsalgoritmen ’random forests’ kan användas för vindkraftsprognoser. Meteorologiska prognoser för vindhastighet, vindriktning, byvindar och luftfuktighet användes. För historisk data testades även vindminimum och temperatur. Resultaten evaluerades utifrån kvadratiskt medelvärde (RMSE), absolut noggrannhet (MAE och NMAE), samt systematiskt fel (NMBE). Resultat från stormen ’Hans’ under 2023 redovisas separat. Historisk data täckte ett år och tre månader. Prognoser täckte en månad. Historisk data användes för att visa att säsong påverkar prognoserna.

De bästa resultaten visade att NMAE minskade (jämfört med en enkel polynomiell modell) från 8.5 % till 6.6 % för historisk data, från 9.5 % till 8.6 % för prognoser under normala förhållanden, och ingen meningsfull skillnad för stormförhållanden. Resultaten indikerar att en random forest-modell kan ge förbättringar av vindkraftsprognoser, samt att vikten av att grunda modellerna på bra data och meteorologiska prognoser med god överensstämmelse sinsemellan är hög.

Abstract

The present thesis investigated using the random forest machine learning algorithm for wind power forecasting. Meteorological prognoses for wind speed, wind direction, gust winds, and humidity were used. For historical data, wind minimum and temperature was also included. The results were evaluated using root means square error (RMSE), mean absolute error (MAE), normalized mean absolute error (NMAE), and normalized mean biased error (NMBE). Results from the 'Hans' storm in 2023 were shown separately. Historical data covered a year and three months. Meteorological prognoses covered a month. Historical data was used to show that seasonality impacts forecasting.

The best results showed NMAE decreasing (compared to a simple polynomial model) from 8.5 % to 6.6 % using historical data, from 9.5 % to 8.6 % using meteorlogical prognoses, and negligible improvements under storm conditions. The results indicate that a random forest model can yield improvements in wind power forecasting. It is simultaneously shown to be important that the models are based on good data and employ good meteorological prognoses with high levels of agreement between them, and with the turbine site.

Till Jeanette, Grabbarna Grus och Rebecca.

Enixe conatus et feliciter gessi,
sed extremo nihil interest.
Mi perendum ut omnia perdam,
sed extremo nihil interest.

Chapter 1

Introduction

Wind power is a steadily growing component of the electricity supply in many countries. For example, in Sweden, wind power grew from 7 % of the electricity supply in 2013 [1, p. 34] to 17 % of the electricity supply in 2020 [2, p. 7].[1] In the EU it represented 15.9 % as of 2022 [3]. Naturally, in order to facilitate wind power in a financially sound manner, companies interested in constructing or currently operating wind turbines rely on forecasting to plan construction and operations. Such forecasting can be done for very long timeframes, analyzing long-term wind power output on any given site to evaluate whether it will be economically feasible to build there, as well as how to site the turbines for maximum output [4]. It can also be done in the short term, in order to inform decisions on the sale of electricity on the spot market; as an example, on the Nord Pool, electricity is traded 24 hours in advance [5]. If supply or demand differs from what was traded as part of this day-ahead trading, the difference is made up for in intra-day trading [6]. Since prices on the intra-day market may differ from those on the day-ahead market, selling more electricity on the day-ahead market than what actually ends up being produced can be a financial liability for a wind turbine operator, potentially reducing their income as the difference has to be made up for either by the operator itself or through an intermediary. This introduces a level of risk, which scales with the uncertainty of the forecasting. At the same time, it is not desirable to undershoot sales either, since an unexpected glut in the market could lead to far lower prices on the intra-day market, thus causing the electricity generated to be sold for much less.

Wind turbine operators therefore have a strong incentive to ensure good short-term forecasts to minimize intra-day trading. Methods for achieving this vary, as do the demands on the forecasts themselves. Operators of large-scale wind farms have the greatest need for robust forecasting, and much work is usually involved in ensuring that this is achieved. This may include an atmospheric flow model based on the local topography combined with statistical modelling [7, p. 62]. For smaller operators of one or a few turbines, however, it may be cost-prohibitive to use such an approach. Instead, a simpler method may be used, procuring meteorological prognoses[2] from a third party

[1] This refers to total production, not installed capacity.

[2] For clarity, meteorological prognoses will be referred to as such from here on, while the term forecast will refer exclusively to wind power forecasting in kW.

and employing statistical methods to correlate production with said forecasts.

At a basic level, this could involve a simple regression correlating power output to wind speed. However, while taking into account other factors such as wind direction could improve such forecasting, it is not always entirely clear how production actually depends on these alternative factors, and conditions may vary between sites. Without taking the topography into account using an atmospheric flow model, it can become difficult to account for this to improve forecasting. Machine learning may provide a feasible way to account for a greater amount of information on meteorological conditions without needing to know the precise nature of any correlation; this would be an economical and time-saving approach to producing sufficiently robust prediction of wind power production, given that suitably accurate meteorological prognoses can be produced.

Thus, for the present thesis, the goal is to investigate how machine learning can be used to improve short-term forecasting for small-scale wind turbine operators. The random forest algorithm was deemed suitable for this, as it has been used successfully for the same purpose in the past [7], and is readily available.

Because this work was undertaken in cooperation with Energy Opticon AB using a relevant case study from an interested customer, emphasis was placed on practical concerns which would aid in the establishmen of a suitable implementation in this specific case, as well as a generalized workflow. Energy Opticon is an IT service provider focusing primarily on forecasting of district heating and cooling, as well as production optimization for heat and electricity, including integration with electricity markets. The relevant case study was for a customer in the Baltic Sea region.

1.1 Goals

To investigate a suitable manner in which the random forest algorithm may be applied to wind power forecasting, five overarching goals were defined:

1. Determine a set of useful input variables (such as wind speed, wind direction, etc.) for a machine learning algorithm so that it produces a suitably robust wind power forecasting model.

It is fairly self-evident that wind power production will depend most heavily on wind speed; strictly speaking, it will depend *only* on *momentary* wind speed *directly incident upon the center of the turbine rotor*. However, meteorological prognoses are typically not provided for the exact location and height of a wind turbine's rotor. Instead, such prognoses will be provided for some nearby location; since both that location and the wind turbine site will be affected by their respective local topographies (such as nearby buildings, hills, mountains, oceans, etc.), it can be assumed that conditions will differ somewhat. Not only that, prognoses for wind speed are typically given as an hourly

average, but variations in wind speed over that time may influence production, and sudden shifts in wind speed or direction may reduce the wind turbine's performance.

All of this is difficult to model, but a machine learning algorithm may be able to account for these variations. Therefore, the goal here is to determine variables besides wind speed which may be suitable in order to improve production forecasting.

2. Investigate in rough terms how the quantity of historical data influences the model's accuracy.

A machine learning model depends on historical data. The question here is what quantity of historical data is actually required; if a smaller quantity of historical data is needed, it simplifies the process and, crucially, means that models can be trained much sooner after installation of a turbine. Conversely, if more data would yield a better model, then retraining the model once more data is available could improve the accuracy of production forecasts, while such efforts may be wasted if the amount of data has already passed a point of diminishing returns.

3. Investigate if sorting by season improves a prediction model's accuracy.

It is possible that, rather than sheer quantity of data, the season from which historical data is drawn has a greater impact on forecast accuracy past a certain point. If this is the case, then building separate models for different seasons, or alternatively retraining the model every so often on the latest available data only, may improve the accuracy of the forecasts. Therefore, it is important to establish if seasonality should be considered when building a wind power forecasting model.

4. Investigate how quickly the error converges depending on parameters in the machine learning algorithm.

The error of a random forest algorithm will usually converge towards some value as a function of the number of decision trees used (see section 2.3). For a low number of decision trees, the model may not perform optimally, while for a high number it will suffer from longer compute time and higher memory usage. It is not expected that computational resources will be at a premium for models in question, but it is nonetheless useful to know how the error behaves when the number of decision trees is increased to ensure that the model performs optimally.

5. Construct a suitable model and investigate the real-world results using actual meteorological forecasts.

Evaluating a model using historical measured data can often yield a good approximation of model performance, but using real-world prognoses will invariably introduce an additional source of potential error in the prediction. This error may take two different forms. On one hand, the prognosis itself may be inaccurate; on the other hand, it may be that the forecast is for a location slightly removed from the place where measured data

was gathered, in which case it may not fully agree with the historical data and cause the model to err. It may therefore occur that the performance of a given model is sufficiently reduced when prognoses are employed that it provides no additional benefit, or performs worse. In particular, if the forecasting for different variables that are employed are less accurate in some cases, it may be less beneficial to use them in a real life scenario. This may be further complicated by the varying types of weather patterns that may eventuate such as storms, where the forecasting may become less reliable due to unpredictable swings in e.g. wind speed.

Therefore, it is essential that the models are tested using real-world meteorological prognoses; what's more, since it cannot be assumed that all variables on which the model is trained will be affected the same, multiple models must be tested in this step using different sets of variables. Furthermore, the actual availability of meteorological prognoses for all the variables that were provided as historical data cannot be assumed, and this must also be accounted for.

Chapter 2

Theory

In the broadest possible sense, there are three ways to approach the type of short-term wind turbine output forecasting which will be of interest to this thesis. First, the physical properties of the turbine itself may be considered in order to build a predictive model based on known constants and specifications [4]. Second, one may employ statistical methods; time series based approaches can be employed, but those usually lose in predictive power beyond a 3-6 hour timespan [7, p. 61]. Lastly, machine learning may be applied to create a forecasting model based on any suitable inputs that are available [8]. In the strictest sense, machine learning algorithms are statistical methods by their nature, but they are usually considered separately from the more 'explicit' forms of statistical methods. This is likely because a machine learning algorithm functions as a black box, as opposed to the less opaque results obtained from a polynomial regression. This has further implications,

These methods are not mutually exclusive. For example, applying statistical methods to a foundational physics-based model may improve its accuracy [7, p. 61]. However, a purely physics based model does have the benefit of being able to operate from day 1 of a turbine's operational life cycle [7, p. 63], whereas a statistical model will only be as accurate as the quantity and quality of the data it is based on.

2.1 Physics based modelling

A physics based approach to modelling wind turbine output generally involves the following steps [7, pp. 61-63]:

- Obtain the actual wind speed and direction at the wind turbine site, and extrapolate the wind speed at hub height (i.e. the height above ground to which the rotor blades are affixed).
- Apply the power curve of the wind turbine to determine its power output (see section 2.1.2).
- If more than one turbine is used, upscale results to the whole wind farm.

The exact process will vary greatly. The first step may involve complex numerical modeling, the second may include statistical modelling, and the third step may not be necessary at all for single turbine installations or be quite complicated for large installations. For installations with capacities exceeding 100 MW, there exists strong financial incentives to improve on the accuracy [9]. However, the focus of the present thesis is on small, predominantly municipal installations, where conditions for the overall wind park will usually extrapolate well with at most some correction for wake loss.

2.1.1 Wind shear

Extrapolating the wind speed at hub height typically involves applying boundary layer theory. At any given point on the map, measured or forecast wind speed is usually given as a simple numerical value, but in actuality it is a function of the height above ground. The Swedish Meteorological and Hydrological Institute (SMHI) for example gives its wind data at a heigh of 10 m above ground [10]. Wind turbines are typically much taller. There are two common methods to account for this phenomenon, which is known as wind shear. The first is the Prandtl log law:

$$U = \frac{u^*}{K} \ln\left(\frac{z}{z_0}\right) \tag{2.1}$$

where U is the wind speed, u^* the friction velocity, $K \approx 4$ the Karman constant, and z_0 the local roughness length [4, p. 15]. If wind speed is given as $U_{measured}$ for a certain height above ground $z_{measured}$ and wind speed U_{rotor} is sought for a different height z_{rotor}, this may be achieved using the following formula derived from the Prandtl log law:

$$U_{rotor} = U_{measured} \frac{\ln(z_{rotor}/z_0)}{\ln(z_{measured}/z_0)} \tag{2.2}$$

The second method is the power law:

$$U = kz^{\alpha} \tag{2.3}$$

where k is a constant, z the height above ground, and α an empirical constant which depends on the terrain. The relationship between wind speeds at different heights can the be derived as [4, p. 17]:

$$U_{rotor} = U_{measured} \left(\frac{z_{rotor}}{z_{measured}}\right)^{\alpha} \tag{2.4}$$

α is not constant with height, but is usually sufficiently close to constant over the height involved for wind turbines that tabulated values can be used, as in Table 2.1 below.

Table 2.1: Approximate values for roughness length and shear index for different terrain types at around 10-80 m above ground [4, p. 16].

Landscape type	Roughness length	Shear index
Water surface (smooth)	0.0002 m	0.08
Short grass, flat open terrain, smooth concrete	0.002 m	0.11
Open agricultural land	0.02 m	0.14
Level country, occasional small trees	0.04 m	0.15
Agricultural land, crops, hedges, some trees	0.10 m	0.18
Agricultural land, distributed buildings and trees	0.2 m	0.20
Villages or small towns, wooded countryside	0.4 m	0.24
Larger towns, tall buildings	0.8 m	0.29
Highly urban landscape, skyscrapers	1.6 m	0.36

The manner in which roughness length affects wind speed as a function of height above ground is demonstrated in Figure 2.1.

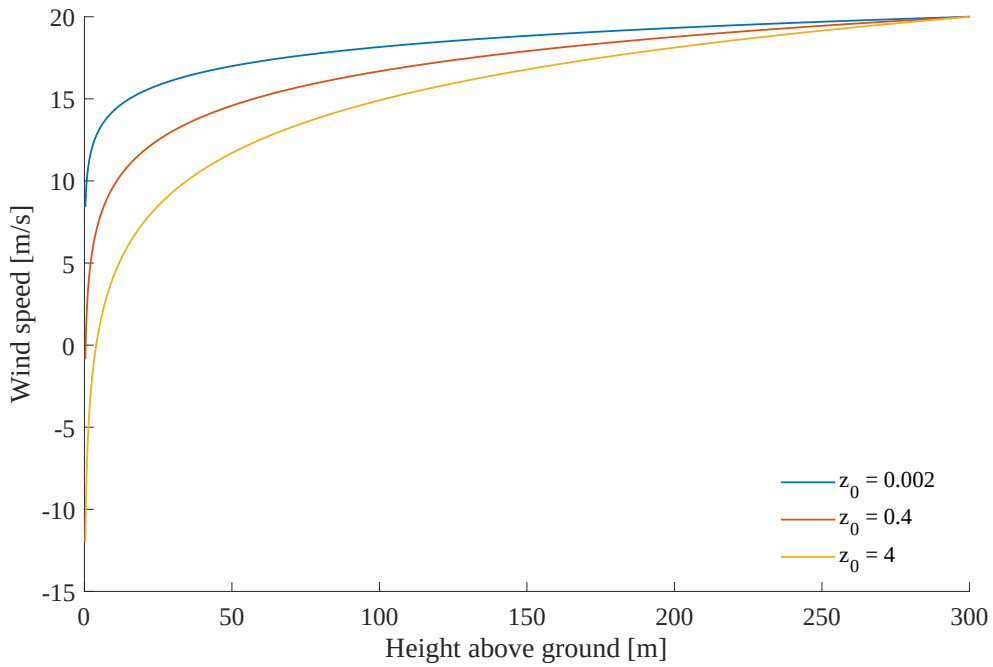

Figure 2.1: How wind speed depends on height above ground for three different roughness lengths z_0 [4, p. 16].

2.1.2 Power curve

The power output from a wind turbine is a function of its incident wind speed, but the specific relationship varies between designs. Most of this lies outside the scope of this thesis, but it is worth pointing out that rotors designed for pitch control have smoother power curves than those designed for stall control. Simply put, a pitch control design controls the pitch angle of the rotor blades to maximize performance and avoid reaching higher speeds than the turbine is designed for; a stall control design cannot do this, but instead has a rotor blade design that stalls above certain rotational speeds and thereby limits its own operations at high wind speeds [4, ch. 4.7].

Wind turbines are usually listed with a cut-in and cut-off speed; this indicates the wind speed at which the turbine begins to produce useful energy, and the wind speed where no further energy is produced. The behavior between these two points is not linear, nor even necessarily constant above the cut-off (especially for pitch control designs). An example of a power curve is shown in Figure 2.2

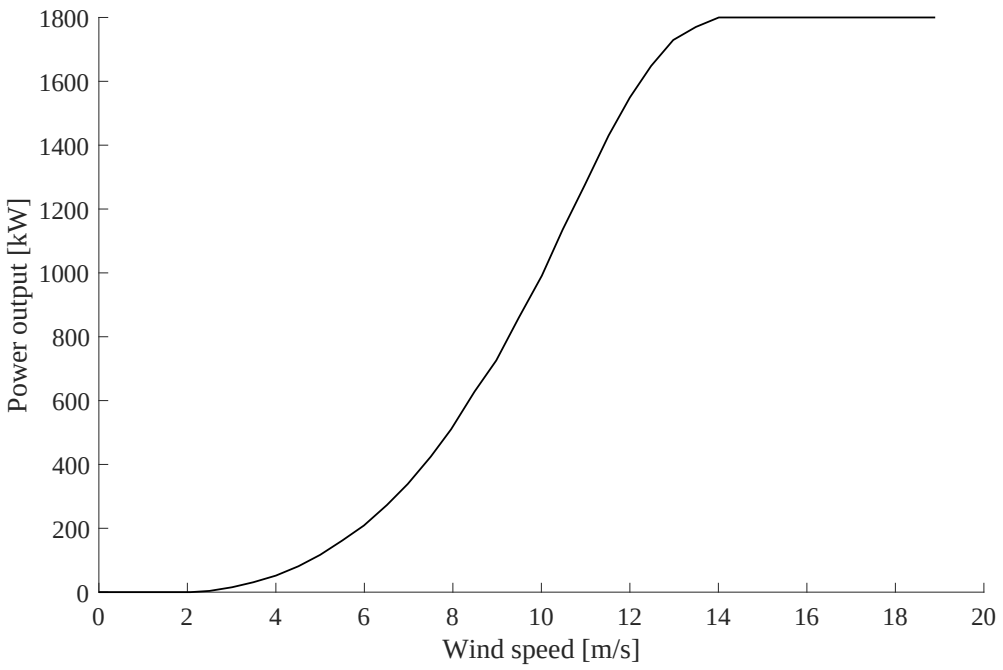

Figure 2.2: An example of a power curve, showing the relationship between wind speed and power output for a wind turbine. This particular example was provided for the turbine north of central site discussed in Chapter 3.

2.1.3 Wake velocity

Besides topographical factors, the actual siting of the wind turbines themselves in relation to each other induces reductions in wind speed and increased turbulence in their wake. If the wind happens to be roughly parallell to two or more wind turbines in sequence, a loss in output will be experienced by the downwind turbines [4, ch. 9.3].

The Jensen-Katic model provides a sufficiently accurate description of these effects. The Jensen-Katic model assumes a conical wake profile. Within the wake cone, the relationship between wake wind speed U_{wake} and the overall site wind speed U_{site} is given by the following:

$$\frac{U_{wake}}{U_{site}} = \left(1 - \sqrt{1 - C_T}\right) D_c \tag{2.5}$$

where D_c is a constant characteristic for the turbine and site, given by:

$$D_c = \left(\frac{D}{D + 2kx}\right)^2 \tag{2.6}$$

where D is the rotor diameter of the turbine that induces the wake, k an empirical constant that depends on the topography (the same as in ch. 2.1.1 above), and x the distance to the next turbine.

For multiple turbines in sequence, the following relationship can be used to find the wake velocity U_N for turbine N in the sequence:

$$\frac{U_N}{U_{site}} = 1 - \sqrt{\sum_{i=1}^{N} \left(1 - \frac{U_i}{U_{site}}\right)^2} \tag{2.7}$$

2.2 Statistical methods - a simple polynomial fit

In principle, statistical methods can be applied on the wind forecast before the power curve is applied, to the power curve itself, or to the final result [7, p. 78]. This may take many different forms; a well constructed model should take into account the behavior of the ouptut as a function of its input in a sensible manner. Fortunately, for the purposes of this thesis, this is not difficult if we consider the wind turbine to be operating in one of three regimes:

1. Wind speed below cut-in $\Rightarrow$ Production is zero
2. Wind speed is between cut-in and cut-out $\Rightarrow$ Production is given by $P = P(U)$

3. Wind speed is above cut-out ⇒ Production is identically the same as installed capacity

For regime 2, P is a function of wind speed U, and can be approximated using a polynomial of suitable order:

$$P(U) = \sum_{i=0}^{N} p_i U^i \tag{2.8}$$

2.3 Machine learning - the random forest algorithm

Machine learning generally denotes algorithms that, taking a set of existing historical data as input, produce a model that can predict outcomes on the basis of similar data. For wind power forecasting, this means that a machine learning model is 'trained' on a set of historical meteorological data paired with production data. If the model is then provided with meteorological prognoses, it should be able to predict power output.

The random forest algorithm is a popular machine learning algorithm, which is widely available as part of statistical and machine learning packages for various script languages. The random forest algorithm is an implementation of decision tree learning which employs bootstrap aggregation to avoid overfitting. These concepts are briefly outlined below.[1]

2.3.1 Decision tree learning

Assume there exists a target parameter y, which for the purposes of this thesis will always be power output. Let x_i be the explanatory variables which y is thought to correlate to - as an example, take x_1 as wind speed and x_2 as wind direction. It is assumed that model y can be modelled as $y = f(x_1, x_2)$.

If y were to have a simple linear or exponential relation to a single input variable, it would be fairly simple to use regression to build a model, but this is not always enough. If y depends upon a single variable x in a manner that is not easily modelled in such a fashion, or if there are multiple explanatory variables where it is not obvious how they might correlate to output y, decision tree learning may be applied as a general method to create the function $f(x_i)$ [11, ch. 9.2].

First, let Z_l denote any sample consisting of y_l and corresponding measured values for $x_{i,l}$. Then let $\{\mathbf{Z}\}_{l=1}^{N}$ be a set of N such samples. Then, consider the cases $y = f_1(x_1)$,

[1] For a direct comparison of several common machine learning algorithms for wind power output forecasting, see Article [8].

$y = f_1(x_2)$, and so on.

Next, for each function f_i, starting at the lowest two values for x_i and moving to the highest two, attempt to set a breakpoint between the values. For all values either side, compute the average or a regression and form the residual sum of squares RSS[11, p. 307] as:

$$\text{RSS} = \sum_{d=1}^{B} (y_d - g(x))^2 \tag{2.9}$$

for the lower side, and vice versa for the upper, where B is the breakpoint and $g(x)$ the average or regression that is fit to the relevant values. Doing this for each possible breakpoint, find the breakpoint for which the lowest RSS is achieved. Once this is done, repeat the process for all parameters x_i, and find the lowest RSS overall. Then, using this breakpoint, form the function:

$$f(x_1) = \begin{cases} g_{lower}(x_1) & \text{for: } x_1 < B \\ g_{upper}(x_1) & \text{for: } B \leq x_1 \end{cases} \tag{2.10}$$

This forms the root of the decision tree, with g_{lower} and g_{upper} forming branches of the tree. By repeating this process, the functions g are broken down into functions depending on either x_1 or x_2 based on what provides the lowest RSS, with any breakpoint resulting in a subset of samples lower than some number not being considered so as to lower the impact of statistical noise on the model. An example of what this might look like is shown in Figure 2.3.

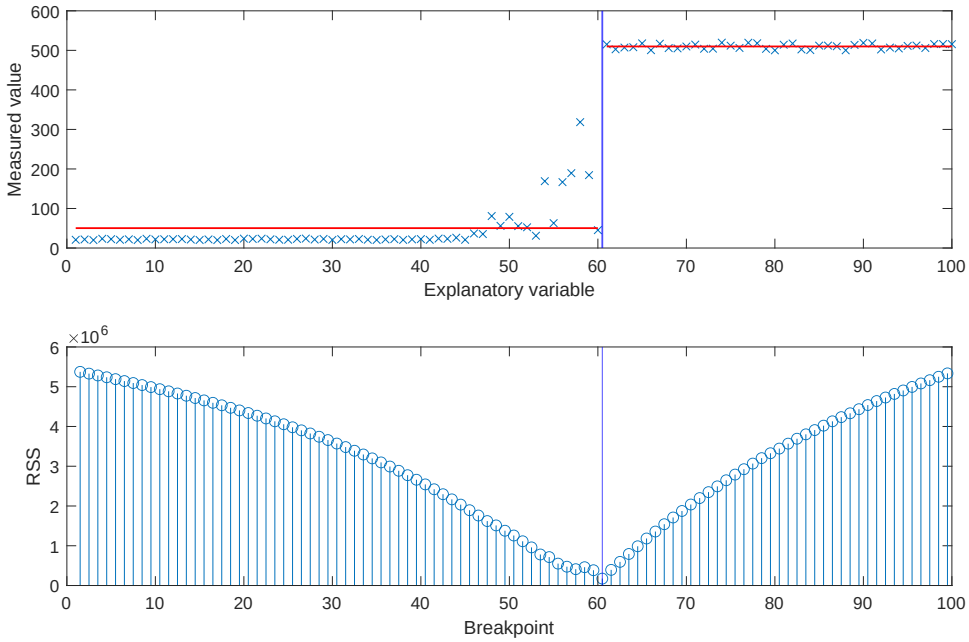

Figure 2.3: A set of generated values as a function of some variable and its corresponding RSS values for every possible breakpoint.

As can be seen, the breakpoint will be placed where the average measured value above and below the breakpoint yield the lowest RSS. Once this is done, the process is repeated - looking at the figure, it should not be difficult to guess that the next breakpoint may end up being somewhere close to 50.

Once this is repeated a suitable number of times (usually until the number of samples is less than a set number), the end result is a decision tree. By considering in sequence the values of the parameters x_i, the decision tree finds a target function $g(x_1, x_2)$ by testing the values against the established breakpoints. Such a target function is called a leaf. Thus, a decision tree is made up of queries such as 'is the variable x_1 lower than b?', connecting via a branch of the decision tree to either another node or a leaf, in the latter case terminating the process.

2.3.2 Bootstrapping

Any time a statistical model is built on some data, there can be a risk of overfitting, and for a decision tree this problem is a major one. Overfitting means that the model fits the data *too* well, finding patterns or correlations in the data that do not actually hold in real life. This can result in a model that appears to perform very well on the sample data, but performs incredibly poorly on any other future data from which output is to be predicted.

Bootstrapping is a means of counteracting this problem. For a set of random samples, bootstrapping works by taking a random sample of said samples. If the original data **Z** consists of N data points, then the bootstrap **Z*** is constructed by taking a number - usually N^2 - of data points at random from **Z** [11, ch. 7.11]. This will usually mean that some data points are left out, and some are duplicated, perhaps multiple times.

If the original data **Z** by random chance happens to show a correlation that doesn't exist, or a stronger or weaker correlation than in reality in some way, this random sampling means that the new sample is unlikely to display the same faulty correlations.

As an example, consider Table 2.2. Here are six observartions of production, and corresponding wind speed, wind direction, and gust winds at some location.

Table 2.2: Example of a small sample of observations.

Obs.	Production (kW)	Wind speed (m/s)	Gust winds (m/s)	Wind direction (°)
1	703.56	3.9	7.7	265
2	771.48	3.9	7.8	264
3	722.16	4	7.8	258
4	662.88	3.1	5.6	266
5	906.00	2.8	6.8	279
6	1236.00	2.8	6.4	280

Creating a bootstrap from this data can be done easily by rolling a six-sided die six times, and inserting the corresponding observation. Take the following sequence of die rolls: 6, 4, 4, 4, 1, 5. The result is Table 2.3 below.

Table 2.3: Example of a bootstrapped sample.

Obs.	Production (kW)	Wind speed (m/s)	Gust winds (m/s)	Wind direction (°)
6	1236.00	2.8	6.4	280
4	662.88	3.1	5.6	266
4	662.88	3.1	5.6	266
4	662.88	3.1	5.6	266
1	703.56	3.9	7.7	265
5	906.00	2.8	6.8	279

Samples 2 and 3 are now not a part of the bootstrapped sample. Samples that are

[2] A lower number may be chosen, e.g. if the quantity of data is such that it would become computationally intensive to use the full amount, particularly if multiple boostraps are to be performed in sequence. If this is done, a corrective factor of $\sqrt{\frac{M}{N}}$ (where M is the smaller sample size) will be introduced. See [12].

outside of the bootstrapped sample are referred to as being 'out-of-bag'. Observation 4 meanwhile is represented three times. Note, however, that although the bootstrapped sample is not genuine, none of the datapoints represented are not. All of the datapoints in the bootstrapped sample are real, it is simply that - in a random process - they are either removed, or duplicated.

2.3.3 Bootstrap aggregation

Bootstrap aggregation, also known as 'bagging' is the process of taking multiple bootstraps and aggregating them to create a better model, in which the overfitting issue can be counteracted by the random sampling of data from the multiple bootstraps employed.

Let as above $\mathbf{Z}$ be a set of data $\mathbf{Z}_1^N$ such that $\mathbf{Z} \in \mathbb{R}^m$, with $m-1$ input variables and one dependent variable. A bootstrapped dataset $\mathbf{Z}^*$ is constructed from $\mathbf{Z}$ by taking at random elements from $\mathbf{Z}$ and placing them in $\mathbf{Z}^*$ until we obtain a set of data $\mathbf{Z}^{*N}_1$; thus, the bootstrap has the same number of elements as the original data. Note that unless the data set is very small, it is virtually certain that $\mathbf{Z}^* \neq \mathbf{Z}$. As a result, some elements in $\mathbf{Z}$ will be excluded from $\mathbf{Z}^*$, and some elements of $\mathbf{Z}$ will be included more than once [11, ch. 7.11].

Then let $S(\mathbf{Z})$ denote a model or prediction formed from $\mathbf{Z}$. Bootstrapping then allows us to obtain an estimate of various aspects of $S(\mathbf{Z})$, such as its variance or the prediction error. [11, p. 250-251] The principal utility of bootstrapping is in assessing the accuracy of an estimation or prediction in such a manner [11, p. 249]; however, a sequence of bootstraps can also be used to improve a statistical model by reducing its variance. This is called bootstrap aggregation, or 'bagging'. [11, p. 282] Let $\hat{f}(x)$ be a predictive model made to fit $\mathbf{Z}$. Let also $\hat{f}^{*b}(x)$ be predictive models fit to bootstrapped datasets $\mathbf{Z}^{*b}$, where $b = 1, 2, ..B$. We can then obtain an estimate using:

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(x) \tag{2.11}$$

It can be shown that as $B \to \infty$, $\hat{f}_{bag}(x)$ approaches the true bagging estimate. [11, p. 282]

2.3.4 Random forests

The random forest algorithm employs the same procedure as bootstrap aggregation, but adds an additional layer by choosing at random one or several explanatory variables at each step.

Random forests are an application of decision tree learning, and are a foundational tool for machine learning. The goal being to create a reasonable prediction of a target value using a set of input values, a random forest is created by making a series of decision trees by employing the bootstrap aggregation procedure but with the additional step of choosing at random one or several explanatory variables at each step. The accuracy of the decision trees are evaluated, and the most accurate examples are continued. The convergence of random forests is shown in, for instance, article [13]. The details are not included here, but it is noted that this means that the problem of overfitting is circumvented when random forests are applied correctly. [3]

The process is as follows (taken from [11, p. 588]):

1. For b = 1 to B:

    (a) Draw a bootstrap sample **Z*** of size N from the training data

    (b) Grow a decision tree T_b to the bootstrapped data by recursively repeating the following steps until the minimum node size n_{min} is reached.

        i. Select m variables at random from the p variables
        ii. Pick the best variable/split-point among the m.
        iii. Split the node into two daughter nodes.

2. Output the ensemble of trees ${T_b}_1^B$

Here, B is the number of times a bootstrap sample is drawn from the training data, and N is the size of each bootstrap sample. p the total number of variables in the training data, meaning if we have wind speed, wind direction, and temperature, $p = 3$. m determines how many variables, chosen at random, should be used at each node (in the previous example, if $m = 2$, then two of those variables would be chosen at random at each node). Different values for m may be tested for.

[3] One example of an overfitting error might be when a higher order polynomial better fits a set of data points, but ends up being a poor predictor of randomly sampled data because it fits the gathered data too well.

2.4 Evaluation of results

A robust set of statistical measures are needed to evaluate the results. Four metrics have been identified as suitable for the present thesis.

2.4.1 RMSE - Root Mean Square Error

RMSE is a commonly used metric for the error. [8, p. 303] It is given by: [14, p. 23]

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2} \tag{2.12}$$

where n is the number of data points, and e_i the difference between each observation and its prediction. RMSE had the simultaneous benefit and drawback of weighting large errors more heavily because the error is squared; this yields a metric more beneficial for determining the behavior of the largest errors when comparing methods.

2.4.2 MAE - Mean Absolute Error

Mean absolute error is given by: [14, p. 24]

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |e_i| \tag{2.13}$$

and yields an intuitive measure of the typical size of the error. MAE contrasts with RMSE in that it weighs all errors equally. In this manner, the two metrics complement each other, yielding a fuller understanding of the performance of a model.

2.4.3 NMAE - Normalized Mean Absolute Error

The normalized mean absolute error, NMAE, is the same as the MAE above, but normalized as a percentage of installed capacity. This enables comparison of different installations, and is commonly used in the literature. [8, p.303] The NMAE is given by:

$$\text{NMAE} = \frac{100\,(\%)}{\text{IP}}\frac{1}{n}\sum_{i=1}^{n} |e_i| \tag{2.14}$$

where IP is the installed power.

2.4.4 NMBE - Normalized Mean Biased Error

The normalized mean biased error yields an indication of whether a model typically over- or underestimates production. The normalization is usually done using the average production, [14, p. 24] but in this case the normalization will be done using installed power IP for the sake of consistency between datasets.[4]

$$\text{NMBE} = \frac{100\,(\%)}{\text{IP}} \frac{\sum_{i=1}^{n} e_i}{n} \tag{2.15}$$

2.4.5 SE - Sine Extremis

It is plausible that the evaluation of a prediction model using each of the above metrics may be affected by the proportion of time that maximum or zero load is predicted.[5] Consider the case where wind speed is much higher or lower than the cut-off or cut-in respectively; the model might then be 100 % accurate for that data point. There is nothing wrong with this inherently, but there are cases where this might become a problem. For instance, in comparing results for two wind turbines, if one of them reaches maximum capacity far more often than the other then the metrics may indicate a much better predictive model simply as a result of the physical conditions at that turbine. The same may occur for different time spans for the same turbine, if wind conditions are substantially different.

As a solution, a 'sine extremis' value will be computed for each metric, whereby all data points where the actual production and forecast are both within a 1 % of the installed capacity from zero or the maximum are removed. In most cases this is not expected to affect results much, but if it were to have a very large impact on the error it would indicate a need to investigate further, while if it does not it corroborates the findings.

[4] This is functionally equivalent to range normalization, whereby the normalization is done using the difference between maximum and minimum output, since for a wind turbine the minimum value will always be zero for suitably long time spans.

[5] This didn't turn out to be the case to any major extent for sufficiently large sets of data, and these values are only shown in the appendix.

Chapter 3

Method

The study was carried out using data collected from a wind turbine in the Baltic Sea region, owned and operated by a utilities company. The company is itself involved in the trading of electricity using the Energy Optima software, and involving other sources of electricity from e.g. combined heat and power plants. The location of the turbine in relation to a central city area and the coastline is shown in figure 3.1. Other key locations relating to where data was gathered from or forecast for are also shown. The turbine has a maximum installed capacity of 1.8 MW.

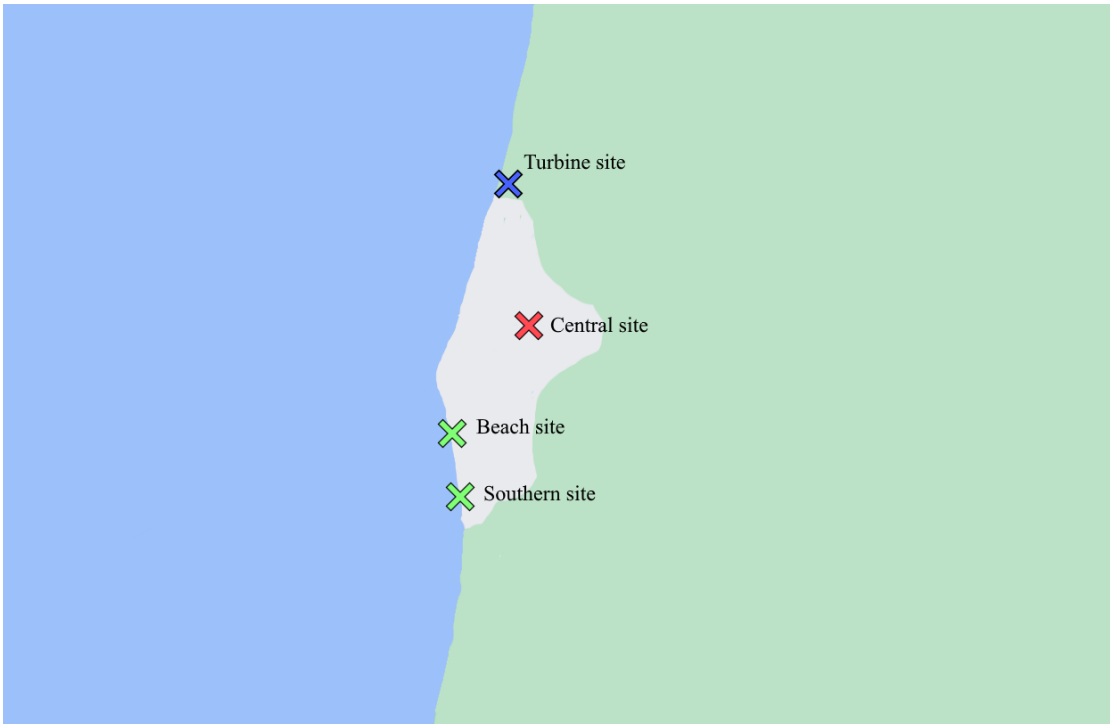

Figure 3.1: Sketch of key locations in proximity to wind turbine site. The turbine is north of a city, situated on the coast.

3.1 Data

For this turbine, production data from 2021-03-01 01:00 through 2022-03-15 00:00, as well as March, May, and June of 2023 was provided. From this, the continuous year running from 2021-03-02 00:00 through 2022-03-01 23:00 was designated as training data, and the data from 2022-03-02 00:00 through 2022-03-14 23:00 was designated av evaluation data together with the data from 2023.

Data was not provided on when the turbines were offline, rather than simply producing no power due to low winds. To account for this, when production was zero for an extended period of time (at minimum ten hours) while wind speeds were such that at least some production would be expected, that data was excluded. 550 hours of data (roughly 23 days) was excluded in this manner from the training data, and 139 hours (roughly 6 days) worth of data was excluded from the evaluation data. Thus, 342 days worth of data was used as training data, and 99 days worth for the evaluation.

Historical meteorological data was procured from LVGMC [15]. The relevant variables were wind speed, wind direction, wind gusts (meaning highest wind speed measured during that hour), wind minimum (meaning lowest wind speed measured in that hour), air temperature, and humidity.

There were two available measurement stations in proximity to the turbine. One was for the city itself, and one for the beach area - see Figure 3.1 above. Both were used. In the case of data from the city proper, there were intermittent gaps in the data for a two week period in May 2023. This data was removed, thus reducing the quantity of data used for the evaluation by a total of about a week.

Forecasts were also drawn from the city itself, as well as for a southern site close to the beach site. Note that no forecasts were available for the beach site, and no measurement station was present precisely at the southern site.

Drawing measurements and forecasts from more than one site was considered important because it was not a given that the closest, central site was the site that would most closely correspond with conditions on the turbine site. The beach and southern sites are both closer to the coast and in areas with comparatively fewer high-rise buildings. Since it has been established that roughness length has a significant impact on wind speed as a function of height above ground (see section 2.1.1), this could plausibly impact how well data from these sites predict conditions at the turbine site.

3.2 Procedure

The process, as outlined in Figure 3.2, begins by obtaining historical wind power production data, and then procuring corresponding historical meteorological observations.

This collection of data is then split into training data and evaluation data. This is because there is always a risk that the model will identify patterns in the training data that do not hold generally, so it is safer to evaluate using data that the model was not trained on.

Models produced in this way are then tested using actual meteorological prognoses, which can then be compared to production data once the time period being forecast is over. Because of the aforementioned factors relating to the day-ahead electricity trading, 48 hours was selected as the cut-off. It is logical to assume the forecasts will be less accurate the further ahead they are, but because the goal is to create forecasts for day-ahead trading in the Nord Pool spot market, forecasts beyond 48 hours are not important.

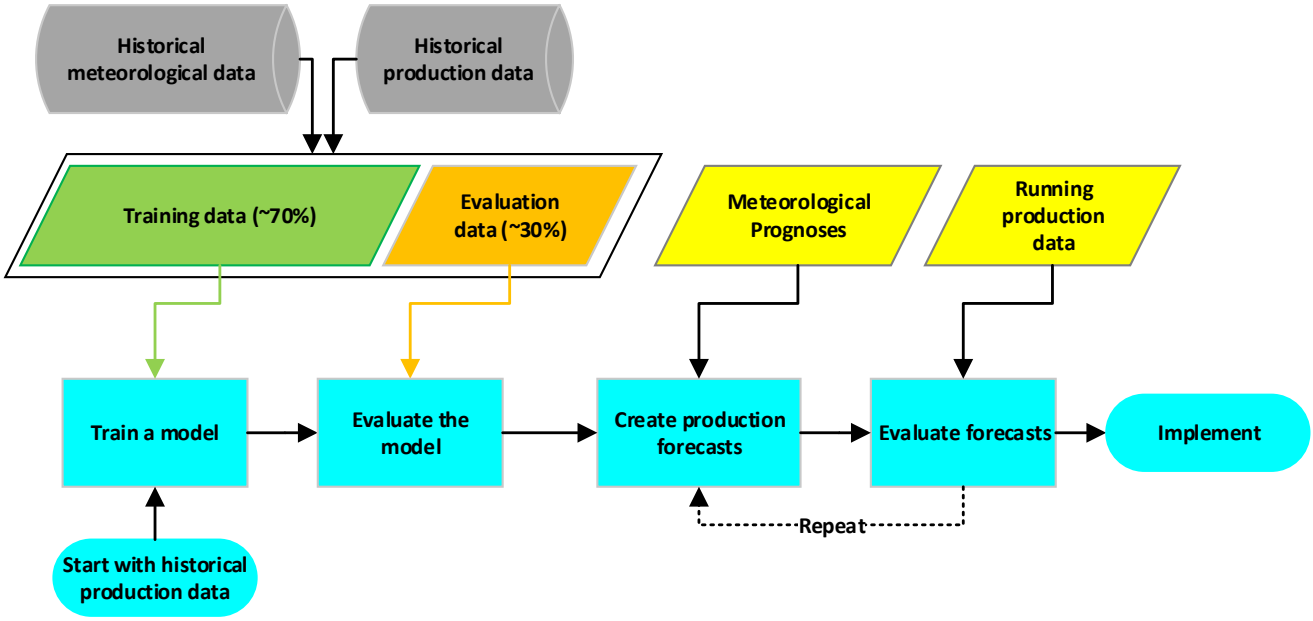

Figure 3.2: Workflow for creating a machine learning model.

The goals of the thesis were, as stated in section 1.1:

1. Determine a set of useful input variables (such as wind speed, wind direction, etc.) for a machine learning algorithm so that it produces a suitably robust wind power forecasting model.
2. Investigate in rough terms how the quantity of historical data influences the model's accuracy.
3. Investigate if sorting by season improves a prediction model's accuracy.

4. Investigate how quickly the error converges depending on parameters in the machine learning algorithm.
5. Construct a suitable model and investigate the real-world results using actual meteorological prognoses.

Recalling that, as discussed in section 2.3 above, a model produced by a machine learning model is typically evaluated using some portion of the historical data, goals 1-4 could be largely addressed as part of this process, with the final step, 5, being an evaluation using actual prognoses. The step by step procedure is described below.

3.2.1 Types of input variables

To determine what types of weather data which improve a machine learning model for predicting wind power output, it was first necessary to create a baseline of performance. Three different models served this purpose. First, a basic physical model, as described in section 2.1 - involving only the extrapolation of wind speed at hub height, and application of the power curve. Second, a basic statistical model using a polynomial fit between cut-in and cut-off values, as discussed in section 2.2. Third, a machine learning model based solely on wind speed.

An attempt was then made to use the output of the two steps involved in the physical model (adjusting wind speed to hub height using the Prandtl log law and applying the power curve), and then training a model on those in order to see if this intermediate step improves forecasting.

After this, the random forest algorithm was used to train a model using each of the additional variables thought to be of use. The full set of variables for which historical data was obtained were:

1. Wind speed (meaning the average wind speed recorded)
2. Gust winds (meaning the maximum wind speed recorded)
3. Wind minimum (meaning the minimum wind speed recorded)
4. Wind direction (in degrees)
5. Relative humidity (as a percentage)
6. Air temperature

All of these were given as hourly values, as was the historical production data.

Once this had been done, models were trained on wind speed as well as the additional variable out of the ones tested that performed the best, then also the second best, and so on until all variables were accounted for.

3.2.2 Quantity of historical data

Since a full calendar year's worth of historical data (offline periods for the turbine notwithstanding) was available to train models on, this data was further subdivided into blocks of 7 days, and models were thus trained on 7 days, 14 days, 21 days, and so on until all data was used. This was repeated for models using only one input variable (wind speed), two, three, and so on in the event that the error would behave differently in each case.

3.2.3 Seasonality of data

Roughly one year and three months' worth of historical production data was provided. Using this quantity of data, it was possible to investigate seasonality since all the data used for evaluation fell into the same 4-month period (March-June). Four groups of data were used: the same 4 months, the 4 months furthest away, and the two 4 month periods immediately preceding and succeeding the March-June period.

3.2.4 Number of decision trees

This step simply involved training models with different numbers of decision trees and seeing how this affected the error. The error was then plotted as a function of the number of trees until a point of diminishing return had eventuated for different sets of variables. This was to establish a minimum number so that the models would not differ from each other by random chance, as might happen if too low a number is chosen, as well as to avoid the unnecessarily long compute-times or high memory usage an excessive number of decision trees might result in.

3.2.5 Forecasts using meteorological prognoses

For the meteorological prognoses, wind direction was only available in an 8-directional format (e.g. N for north, NW for north west, etc.) To circumvent this issue, wind direction was converted to degrees in 45° increments. Wind minimum was not available, and so was not included. Temperature was available, but was nonetheless excluded after consideration of results up to that point. A total of 27 forecast periods of 48 hours each were used, running from August 7th through September 10th. Prognoses were taken from the central site and from the southern site.

August 7th to 12th 2023 marked a period of unusually inclement and unpredictable weather in the Baltic Sea region [16]. This presented an uncommonly good opportunity to investigate how forecasting models perform under such conditions. Therefore, data

which was gathered for the duration of the event is shown both separately from the remaining data, and together.

Chapter 4

Results and discussion

4.1 Univariate models

The error metrics for the univariate models based on meteorological data from the beach site and the central site are shown in Tables 4.1-4.2 below. The physical models simply show what results from applying the Prandtl log law and the power curve, as discussed in section 2.1. The polynomial model is a fourth order polynomial with cut-in and cut-out considered, as discussed in section 2.2. The Wind speed only model is a random forest model trained on wind speed. The last two models are also random forest models, trained on wind speed adjusted using the Prandtl log law and the output of the physical model respectively.

Table 4.1: Univariate models for the beach site.

Variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
n/a - Physical model	343	227	12.5	-12.1
n/a - Polynomial	220	154	8.5	0.2
1 - Wind speed only	221	156	8.6	0.9
1 - Wind speed, adjusted	221	157	8.7	0.9
1 - Physical model output	221	158	8.8	1.0

Table 4.2: Univariate models for the central site.

Variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
n/a - Physical model	556	359	19.9	-19.9
n/a - Polynomial	284	205	11.3	0.6
1 - Wind speed only	277	203	11.2	2.2
1 - Wind speed, adjusted	277	203	11.3	2.3
1 - Physical model output	278	208	11.5	2.5

The physical model does not, on its own, perform well. This is expected, since it is a very primitive model, and no account is taken of the fact that wind speed will differ on the turbine site compared to where measurements are made.

The polynomial model and the random forest model trained on wind speed only perform similarly in both cases. This is also expected; a random forest model trained on a single variable should logically result in a similar outcome to a manually constructed polynomial fit. The only difference of note is that the polynomial model outperforms the univariate RF model in the NMBE metric. This means that it is less likely to overestimate production. Other differences are minor, although RMSE does improve slightly for the random forest model for the central site data specifically.

This indicates that when only one variable is available, a manually constructed polynomial fit that takes cut-in and cut-out into account will perform similarly to a random forest model, and may even outperform it slightly.

It was hoped that using components of the physical model as intermediate steps may improve performance, but this was not the case. The possibility cannot be completely disregarded that, had nacelle wind speed measurements been available, that may have improved results as an intermediate step, by using RF to predict wind speed and then applying the power curve, instead of using RF to predict production directly. It is nonetheless deemed unlikely, since the primary benefit of such an intermediate step would be that the power curve is taken into account separately and with a higher level of accuracy; this is still done in the above case, and yet the error is not reduced.

4.2 Bivariate models

Tables 4.3-4.4 below show the error metrics for bivariate RF models, where models are trained using wind speed as well as one other variable. The polynomial and univariate RF model using wind speed only are shown for comparison.

Table 4.3: Bivariate models for the beach site.

No. variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
n/a - Polynomial	220	154	8.5	0.2
1 - Wind speed only	221	156	8.6	0.9
2 - Gust winds	204	138	7.7	0.0
2 - Wind direction	221	143	8.0	-0.1
2 - Wind minimum	221	144	8.0	-0.0
2 - Humidity	221	159	8.8	1.0
2 - Air temperature	222	158	8.8	1.3

Table 4.4: Bivariate models for central site.

No. variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
n/a - Polynomial	284	205	11.3	0.6
1 - Wind speed only	277	203	11.2	2.2
2 - Gust winds	247	180	10.0	2.0
2 - Wind direction	244	174	9.6	1.9
2 - Wind minimum	245	174	9.6	1.9
2 - Humidity	279	201	11.2	1.9
2 - Air temperature	280	203	11.3	2.4

Of the variables tested, gust winds, wind direction, and wind minimum were effective in reducing the error. Humidity and air temperature were not.

In this case, the behavior differed between the two data sets. For the beach site, the bivariate models with gust winds, wind direction, and wind minimum all had very low NMBE, indicating that they neither over- nor underestimated production. This was not the case for the data from the central site, where the NMBE values did not improve as much. Additionally, only the bivariate model using gust winds achieved an improvement in RMSE for the beach site, but all three did for the central site, and by a much wider margin.

Since RMSE is weighted towards outliers, this indicates that using these input variables can reduce outliers for models based on weather data which differs more from turbine site conditions, but that this improvement may be less if the underlying data is already close to site conditions.

4.3 Sequence of multivariate models

Tables 4.5-4.6 show the error when the input variables discussed above are added in sequence. Thus, the first model is wind speed only, then gust winds is added as a second variable, then wind direction as a third, and so on. This is done roughly in the order of efficacy the variables showed for the bivariate models. Air temperature and humidity are both added as a fifth variable, and then together as fifth and sixth, in case they provide an improvement.

Table 4.5: Sequence of multivariate models trained on beach site data.

No. variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
n/a - Polynomial	220	154	8.5	0.2
1 - Wind speed only	221	156	8.6	0.9
2 - Gust winds added	203	138	7.6	0.0
3 - Wind direction added	202	130	7.2	0.0
4 - Wind minimum added	194	126	7.0	-0.3
5 - Air temperature added	184	124	6.9	-0.2
5 - Humidity added	183	119	6.6	-0.5
6 - All variables	180	121	6.7	-0.3

Table 4.6: Sequence of multivariate models trained on central site data.

No. variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
n/a - Polynomial	284	205	11.3	0.6
1 - Wind speed only	277	203	11.2	2.2
2 - Gust winds added	245	179	9.9	2.1
3 - Wind direction added	210	151	8.3	1.6
4 - Wind minimum added	208	147	8.1	1.5
5 - Air temperature added	204	146	8.0	1.5
5 - Humidity added	197	139	7.7	1.2
6 - All variables	197	141	7.8	1.5

In most metrics, it is fairly clear that the error is reduced for each variable up to and including wind minimum. The exception is NMBE, which for the beach site is lowest in the bivariate model. This may simply be an artefact of the specific conditions in the dataset used, since the NMBE is already very low for these models relative to NMAE.

Both sets of data show very similar results, with the beach site seemingly the better set of data overall, likely indicating that this data is from a measurement station that more closely correlates with conditions for the turbine. This does not mean that the beach site data is the most useful, as that depends also on which set of data most closely corresponds with the meteorological prognoses.

All models outperform the basic physical model by a wide margin. Even the worst models never ended up with an NMAE of more than 11.5 %, compared to 19.9 % for the physical model. The best results were obtained using all variables except temperature from the beach site data, yielding an NMAE of 6.55 % and an NMAE of 119 kW.

Of the variables investigated, wind speed, wind direction, gust winds, and wind minimum consistently improved results. For air temperature and humidity, results were very

inconsistent. When added as a second variable, neither improved results. Yet, added as a fifth and sixth variable, results were improved. It is difficult to say if this is an artefact of the dataset.

It is particularly interesting that MAE and RMSE are not always reduced by the same rate, implying that the behavior of the larger errors differs from the overall average. Consider that a model trained on wind speed and wind direction for the beach site data will yield an improved MAE value compared to wind speed alone (going from 156 yo 143), but the RMSE stays the same at 221. This implies that the behavior of outliers does not change much, and may even be worsened somewhat.

It appears plausible that this results from the much stronger correlation that production will have with wind speed than with any other variable. Wind direction, then, improves overall accuracy, but does not meaningfully affect the outliers that are primarily the result of the wind speed variable.

Notably, gust winds does appear to reduce RMSE; applying gust winds and wind speed moves RMSE from 221 to 204.

It is in general difficult to determine why the results for each of these models behaved in the way that they did. Machine learning algorithms are by their nature not explicit in the way that a typical statistical model is, functioning effectively as a black box, and thus it is not clear why the results differ here. What is very clear is that care must be taken in producing a predictive model. The interactions between results and the input variables is not necessarily straightforward, as a result of which active user input is required as part of an iterative process. It cannot simply be automated as a 'fire-and-forget' process using any data that seems like it might be helpful. The clearest example of this is the inconsistent behavior from adding the humidity variable.

At the same time it seems clear that more data is usually better, with the caveat that this will of course only apply if the historical weather data is closely matched to the actual meteorological prognoses. It also appears that using RF models has a greater overall impact on results for less accurate meteorological data; improvements from using an RF model on data from the central site were much greater, particularly on RMSE, in absolute terms; improvements were also more consistent. Still, this was only barely able to compensate for the inferiority of the central site data in predicting conditions on the turbine site. The worst case for the beach site, after all, showed an NMAE of 8.6 %, while the best case for the central site showed 7.7 %. Thus, it appears that procuring more reliable sources of data is more important than improving the modelling.

Regardless of the efficacy of humidity and temperature as explanatory variables, it does seem that wind speed, wind direction, gust winds, and wind minimum are desirable for creating a viable wind forecast prediction model using machine learning.

It is also notable that excluding the values close to maximum and minimum did not impact the results to any major extent. There was a tendency to overestimate production

for the single variable models for the beach site data, as well as the two variable models using air temperature or humidity, with the NMBE being above 1 % in all those cases. For the central site data, there was a much stronger tendency to overestimate production, with the NMBE never below 1.1, and as high as 2.3 for the single variable model. This tendency towards overestimation is a slight concern, since over longer time spans it could lead to an overestimation of how much power can be delivered to the grid; however, a typical overestimation in the order of the lower tens of kW is acceptable.

4.4 Quantity of historical data

Figures 4.4-4.5 show how the NMAE depends on the number of weeks the RF models are trained on.

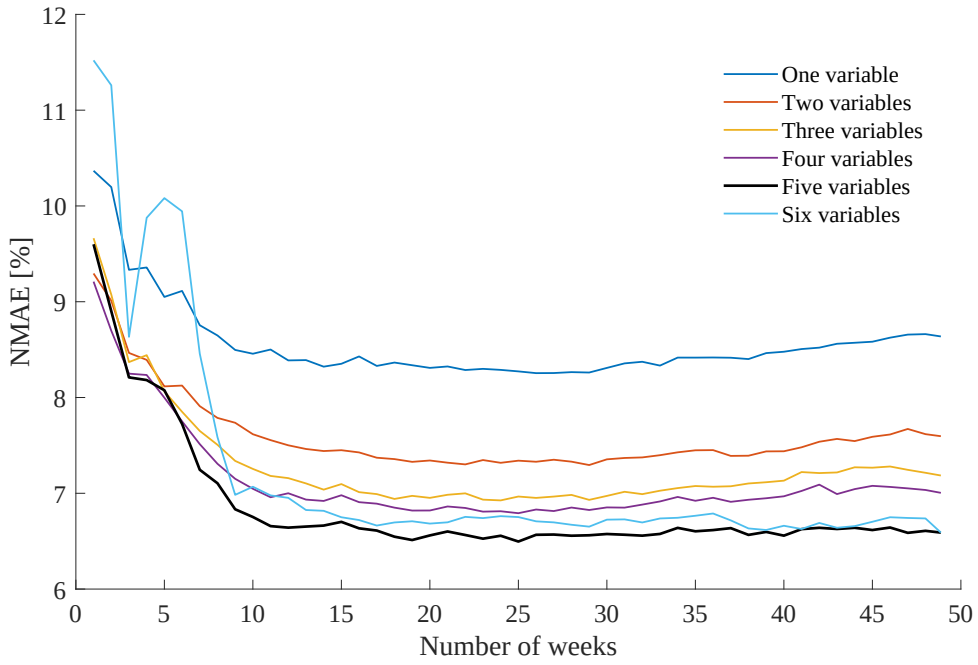

Figure 4.1: How NMAE depends on the number of weeks in the input data for models based on different sets of variables, using beach site data. The best model, with five variables, is shown as a bold line.

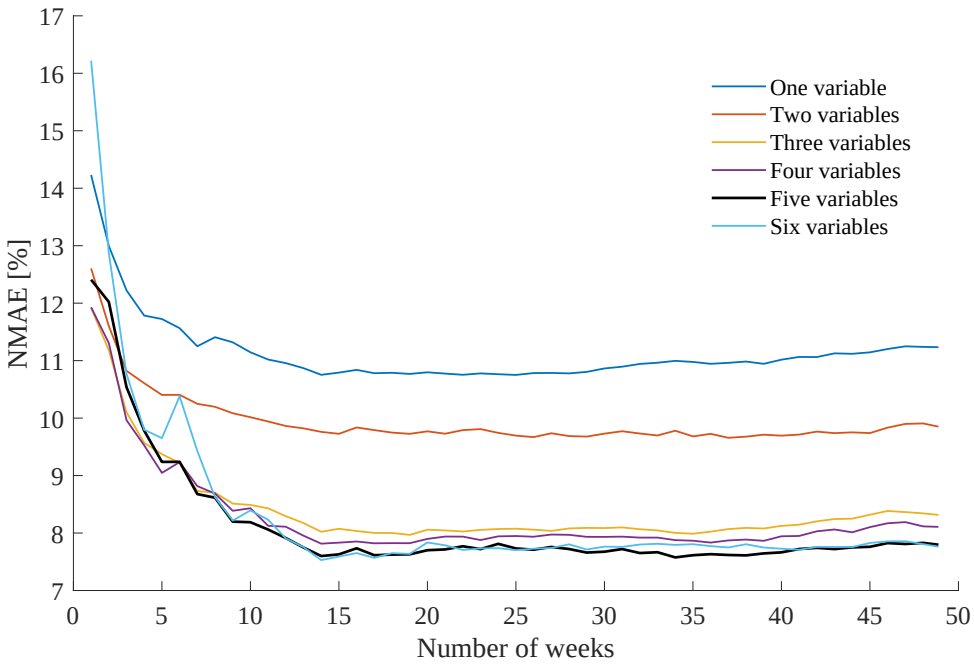

Figure 4.2: How NMAE depends on the number of weeks in the input data for models based on different sets of variables, using central site data. The best model, with five variables, is shown as a bold line.

The results seem to indicate that the additional benefit of more data tapers off somewhat by around 15 weeks. Interestingly, results seem to worsen slightly with more weeks. This may be a result of the nature of the data. If models had been trained on a full year's worth of data and then evaluated on another full year, one would certainly expect that using the full amount of data would be better. That this isn't the case, coupled with the fact that the evaluation data is from a relatively small timespan in spring and summer, may indicate that there is at least some dependency on seasonality. This appears to be most pronounced for the beach site data (which is in general the strongest predictor of production output), and more so the fewer variables are used. A tentative suggestion may be that using more input variables might account for the difference in seasonality, and in lieu of this accounting for seasonality could account for having fewer input variables.

4.5 Seasonality

To investigate if seasonality has an impact on results, models were trained on different sets of data blocked by time of year. Because all of the evaluation data was contained within the months of March, May, and June, the base models were based on data from the four months of March through June. This was compared to models trained using

data from September through December (the four month period furthest away from the evaluation data), as well as data from November through February and July through October (the period immediately before and after.) It was hoped that the quantities of data would be similar, but when offline periods were accounted for, the total number of days for each were:

Table 4.7: Number of days in each seasonal period.

Four month period	No. of days in data	Approx. No. of weeks in data
March-June	117 days	17 weeks
July-October	117 days	17 weeks
September-December	155 days	22 weeks
November-February	107 days	15 weeks

This is not considered likely to be a major influence on results. Figures 4.4-4.5 seem to indicate that the additional benefit of more data tapers off somewhat by around 15 weeks, and all the periods above have at least that much data in them.

Tables 4.8-4.11 show results for the different models when grouping data in the manner described above.

Table 4.8: Statistical measures of the error for models trained on data for the beach site using data from March through June.

No. variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
1 - Wind speed only	220	151	8.3	-0.1
2 - Gust winds added	204	134	7.4	-0.4
3 - Wind direction added	198	128	7.0	-0.4
4 - Wind minimum added	195	125	6.9	-0.4
5 - Humidity added	184	119	6.6	-0.5
6 - All variables	184	122	6.7	-0.5

Table 4.9: Statistical measures of the error for models trained on data for the beach site using data from July through October.

No. variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
1 - Wind speed only	229	156	8.6	-0.8
2 - Gust winds added	209	138	7.6	-1.8
3 - Wind direction added	219	136	7.5	-2.5
4 - Wind minimum added	217	135	7.4	-2.3
5 - Humidity added	209	131	7.2	-2.3
6 - All variables	204	138	7.6	-0.4

Table 4.10: Statistical measures of the error for models trained on data for the beach site using data from September through December.

No. variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
1 - Wind speed only	227	163	9.0	1.4
2 - Gust winds added	208	142	7.8	-0.4
3 - Wind direction added	223	140	7.8	-1.5
4 - Wind minimum added	215	137	7.6	-1.4
5 - Humidity added	200	132	7.3	-0.9
6 - All variables	196	137	7.6	0.1

Table 4.11: Statistical measures of the error for models trained on data for the beach site using data from November through February.

No. variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
1 - Wind speed only	243	181	10.0	4.3
2 - Gust winds added	222	159	8.8	3.0
3 - Wind direction added	223	160	8.8	2.8
4 - Wind minimum added	213	151	8.3	2.5
5 - Humidity added	199	143	7.9	2.4
6 - All variables	191	143	7.9	3.1

Results are consistently superior for the data that best corresponds to the evaluation data. A reasonable point of comparison seems to be the NMAE for the 5 variable models; for March-June, it is qty6.6%. For July-October, it is 7.2 %. For September-December, it is 7.3 %. The worst results are obtained for November-February, being 7.9 %. Of particular note is that the performance of models with fewer variables using the March-June dataset sometimes outperform models with more variables using other datasets. As an example, the NMAE for a single variable model based on data from March-June was 8.3 %, while for two variables in November-February it was 8.8 %. It is also interesting to note that the NMBE tends to worsen substantially for datasets based different months than the evaluation data. In particular, the November-December dataset yields models that overestimate production by between 2.4 % and 4.3 %, while the March-June dataset at worst underestimates production by 0.5 %.

A comparison of MAE, NMAE, and NMBE for all the datasets (including the models trained on a full year's worth of data) is given in Table 6.3 below.

Table 4.12: Statistical measures of the error for models trained on different datasets separated by month for the beach site.

No. variables - Description	Data grouping	NMAE (%)	NMBE (%)
1 - Wind speed only	Full year	8.6	0.9
	March-June	8.3	-0.1
	July-October	8.6	-0.8
	September-December	9.0	1.3
	November-February	10.0	4.3
2 - Gust winds added	Full year	7.6	-0.0
	March-June	7.4	-0.4
	July-October	7.6	-1.8
	September-December	7.8	-0.4
	November-February	8.8	3.0
3 - Wind direction added	Full year	7.2	-0.4
	March-June	7.0	-0.4
	July-October	7.5	-2.5
	September-December	7.8	-1.5
	November-February	8.8	2.8
4 - Wind minimum added	Full year	7.0	-0.3
	March-June	6.9	-0.4
	July-October	7.4	-2.3
	September-December	7.6	-1.4
	November-February	8.3	2.5
5 - Humidity added	Full year	6.6	-0.5
	March-June	6.6	-0.5
	July-October	7.2	-2.3
	September-December	7.3	-0.9
	November-February	7.9	2.4
6 - All variables	Full year	6.7	-0.3
	March-June	6.7	-0.5
	July-October	7.6	-0.4
	September-December	7.6	0.1
	November-February	7.9	3.1

It seems apparent that using seasonally appropriate data can have an impact on results. However, it also appears that training a model on a full year's worth of data is not much worse than training on only seasonally appropriate data. What is very clear is that training a model on data from only part of the year will yield unreliable results when used for data from other times of the year.

What this analysis does not make clear is whether a model trained on the same four-month period from multiple years would yield improved performance. Nor does it make clear

what the exact separation should be. The quantity of data available was insufficient for doing so. Nonetheless, it indicates that grouping data by season may be an option for improving performance of such models.

It is, however, reasonable to suggest from the available information that the otherwise more convenient option of automatically updating the model using the latest 3-4 months on an ongoing basis may not work well. This is because, out of the four-month groupings, it was not the one immediately preceding the four month grouping the evaluation data belonged to that was the second best. Indeed, it was actually the one which performed the worst. This suggests that if seasonality should be accounted for, this should be done with a more fixed blocking of the year into seasonal categories. The specifics of how this would be done is not possible to determine using the available data, but it is plausible that this may vary depending on the site.

Regardless, the improvement provided by using seasonally appropriate data over a full year is not sufficiently substantive to indicate that it is worthwhile. While never worse, the March-June models are at best only 0.3 points better in NMAE, and it is difficult to determine if the differences in NMBE are actually significant.

One interesting thing to note is that there does appear to be a tendency for the July-October models to underestimate production in March-June, and for November-February models to overestimate production. Particularly for the latter case, where the NMBE is in the range 2.4 % to 4.3 %.

While it is not clear exactly what these seasonal differences may be, wind speed and direction may be considered plausible culprits. A wind rose published by LVGMC, shown in Figure 4.3 below, indicates certain seasonal differences in wind direction and intensity (in the Beaufort scale, higher numbers correspond to higher winds.) These differences appear to be both in regards to wind speed and direction - note the higher proportion of lower wind speeds spring and summer compared to winter, and the dofferences in how wind direction is distributed.

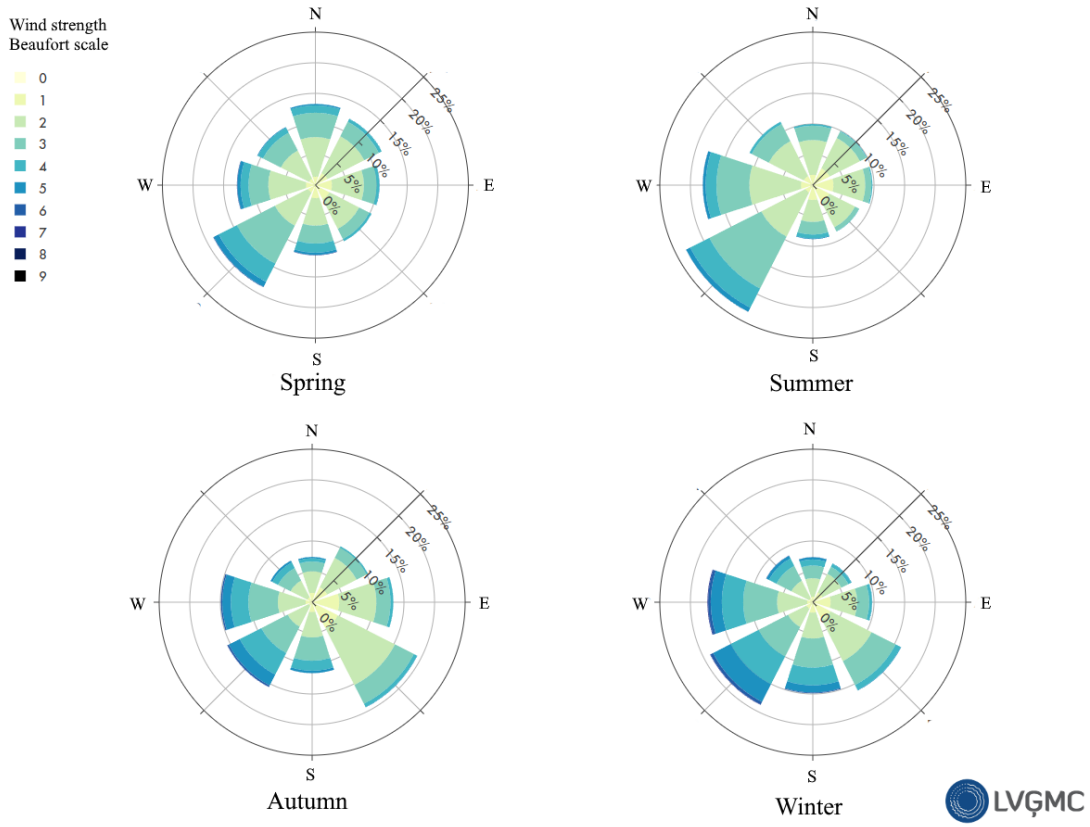

Figure 4.3: Wind rose for the relevant area. [17]

4.6 Number of decision trees

For a random forest algorithm, the main parameter that can be adjusted is the number of decision trees. The impact on the overall error of the model of adjusting this is shown in figures 4.4-4.5.

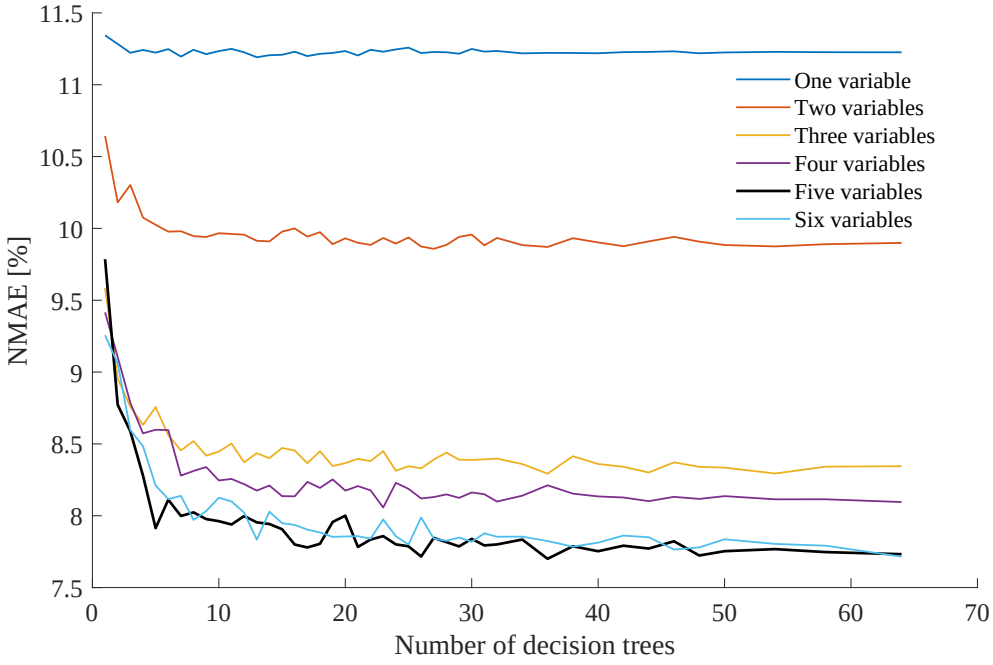

Figure 4.4: How NMAE depends on the number of decision trees for models based on different sets of variables, using central site data. The best model, with five variables, is shown as a bold line.

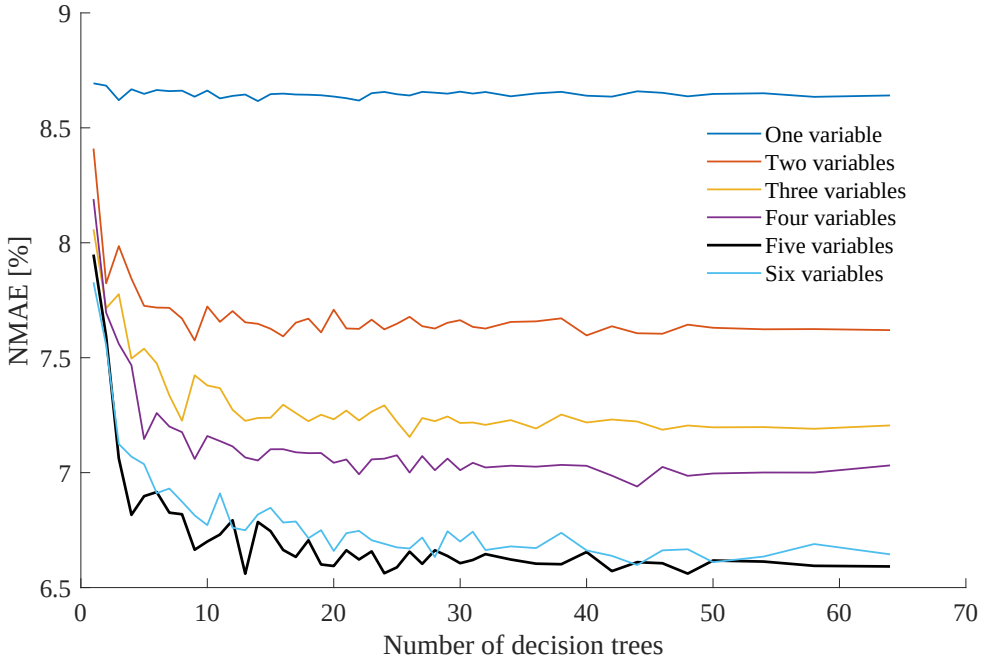

Figure 4.5: How NMAE depends on the number of decision trees for models based on different sets of variables, using beach site data. The best model, with five variables, is shown as a bold line.

As can be seen, when using only one explanatory variable, no improvement is obtained when using more decision trees. This is entirely expected, since using only one variable will result in little more than a basic regressive model for which a random forest algorithm is unnecessary. As more variables are added, the effect of increasing the number of decision trees is magnified up to a point.

It is also not surprising that improvements are most stark while increasing the number of trees up to the amount of variables used, with more incremental improvements past that point.

Using additional decision trees can become computationally intensive, but the number of trees used in this model is still sufficiently low that very little is to be gained from reducing the number.[1] The assumption is that 64 decision trees should be more than sufficient to ensure consistent results without resulting in unacceptably long compute times.

[1] Compute time for a single model typically did not exceed 1 minute on a mid-range PC from around 2015

4.7 Results for the 'Hans' weather event

Error measured for forecasts during the Hans weather event are shown in Tables 4.13-4.14. Models were all trained on beach site data; although it was thought that models trained on data from the central site may perform better for prognoses from the central site, this did not turn out to be the case.

Table 4.13: Errors for forecasts during Hans using weather forecast from the central site.

No. variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
n/a - Polynomial	388	276	15.3	-2.1
1 - Wind Speed	398	279	15.5	-2.0
2 - Wind Gust	399	284	15.8	-0.7
2 - Wind direction	395	285	15.8	-2.0
2 - Humidity	400	282	15.7	-2.9
3 - Wind gust and direction	375	262	14.5	-2.3
4 - Humidity added	369	258	14.3	-3.0

Table 4.14: Errors for forecasts during Hans using weather forecast from the southern site.

No. variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
n/a - Polynomial	324	234	13.0	-0.8
1 - Wind Speed	335	238	13.2	-1.6
2 - Wind Gust	338	246	13.7	-0.1
2 - Wind direction	341	244	13.6	-2.1
2 - Humidity	324	232	12.9	-2.1
3 - Wind gust and direction	318	228	12.7	-1.7
4 - Humidity added	328	236	13.0	-3.2

There is obviously a lot of uncertainty in these results due to the very small sample size. Nonetheless, it is observed that machine learning based models cannot be stated to have consistently outperform the simple polynomial model. Any differences observed between models may be due to random chance. In particular, very little can be said of NMBE with the very small quantity of samples. The difference between the two locations for which meteorological prognoses were obtained, however, was large.

This is a very important result. the southern site is further away from the turbine site, but because weather conditions in the southern site presumably match more closely to conditions on the turbine site (and/or perhaps to conditions for the measurement station from which historical weather data was obtained), the resultant production forecasts were better. This indicates that much care must be taken in choosing a location from

which forecasts are sourced. It is not sufficient to simply choose the option closest to the turbine site.

One caveat to these results is the format in which wind direction was provided. Because the format was 8-directional, forecasts using wind direction may have suffered.

The southern site forecasts for the polynomial, bivariate (with gust winds), trivariate, and quadrivariate models are shown in figures 4.6-4.9. Because of the manner in which the data was gathered, the forecasts overlap.

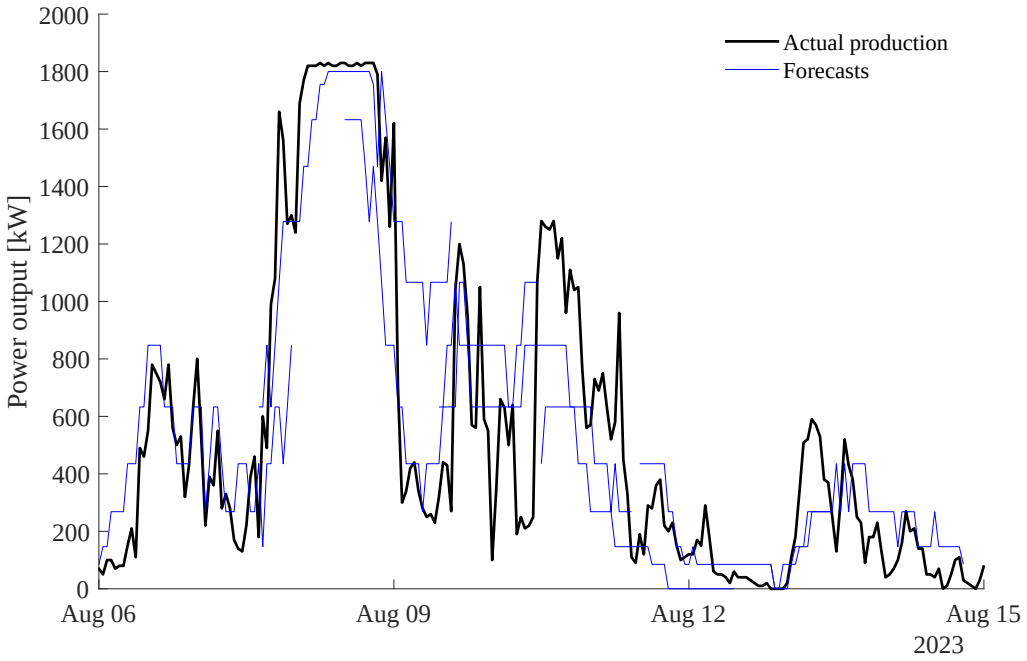

Figure 4.6: Forecasts using the polynomial model compared to actual production during the Hans weather event.

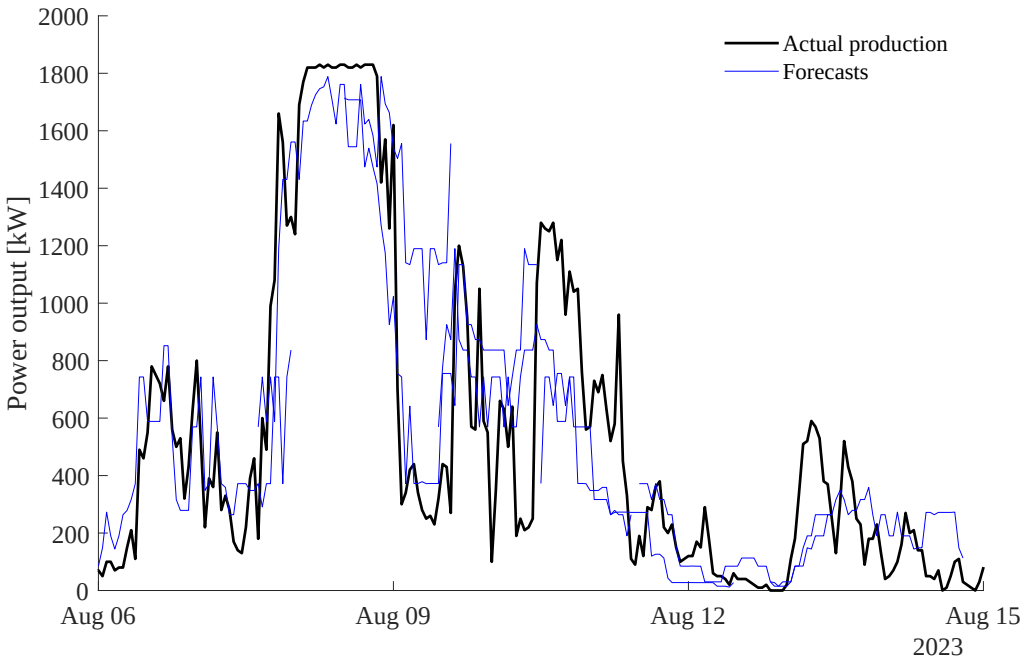

Figure 4.7: Forecasts using the bivariate model with gust winds compared to actual production during the Hans weather event.

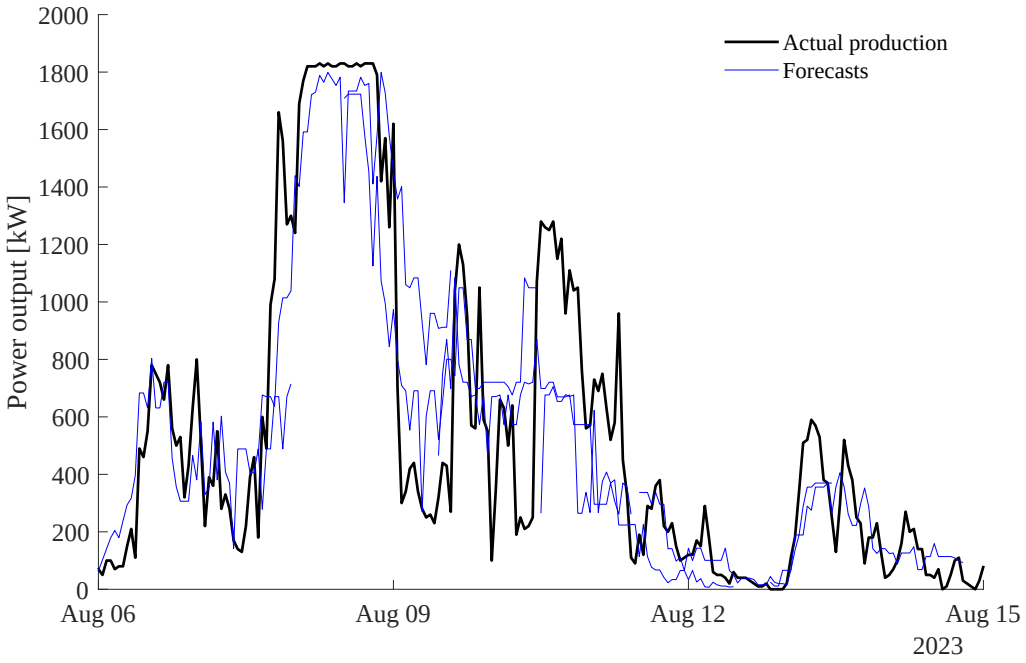

Figure 4.8: Forecasts using the trivaraite compared to actual production during the Hans weather event.

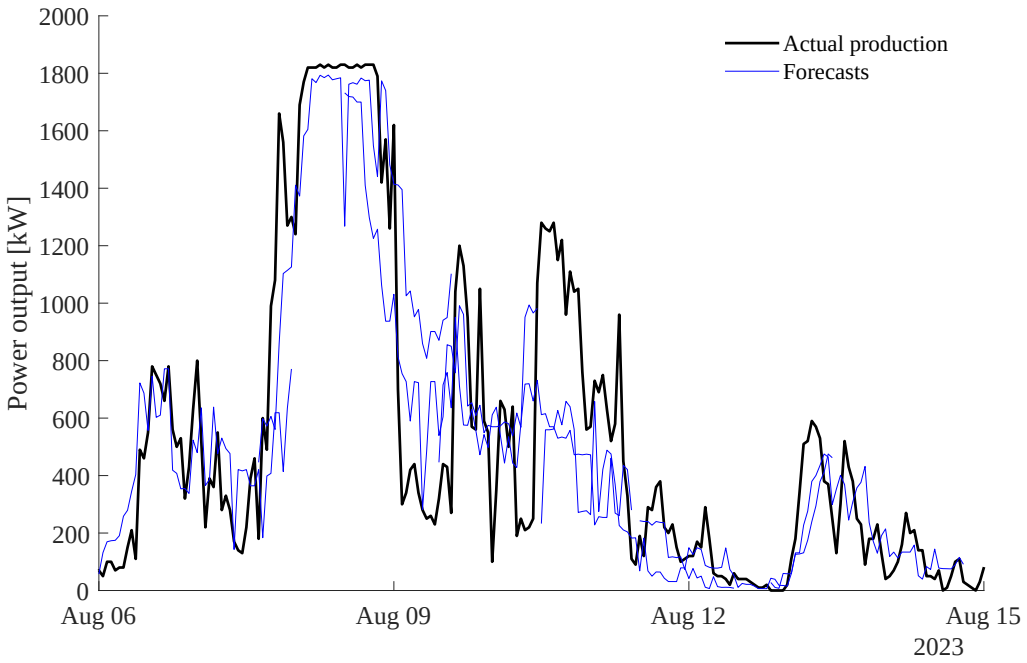

Figure 4.9: Forecasts using the quadrivariate model compared to actual production during the Hans weather event.

Certain features of these plots may be considered noteworthy. For example, at 14:00 on August 9th, while actual production was only 270 kW (although it would quickly rise to 1'200 kW), the polynomial model predicted 1278 kW, the bivariate model predicted 1556 kW, the trivariate model predicted 1110 kW, and the quadrivariate model predicted 1102 kW. This is just one of many potentially noteworthy features, more of which will be discussed below, but it is noteworthy for the large increase in the error by the bivariate model, while the trivariate and quadrivariate models showed an improvement.

It is also visually apparent that the multivariate models performed worse on August 9th. Where production showed a trough of around 300 kW, the trivariate and quadrivariate models in particular predicted higher production. This is made clear by simply looking at the lower of the forecasted values in Figures 4.8 and 4.9 for August 9th[2]. Notice how the trough is wider in the forecasts than actual production would indicate. Now compare this to Figures 4.6 and 4.6. These plots are much closer to the actual production in this case.

There are also a few other interesting observations, such as the sudden and inexplicable underestimate of production in the tri- and quadrivariate models on August 8th (where actual production was consistently around 1'800 kW). This could indicate a difficulty

[2] Again, the reader is reminded that forecasts may overlap, since meteorological prognoses for 48 hours periods were collected roughly once every 24 hours. It is the lowermost values that are of interest in this case.

for the machine learning algorithm to handle values in the vicinity of the maximum production.

4.8 Normal weeks

Results for all weeks considered outside of the Hans storm are shown in Tables 4.15 and 4.16 below. These values are for 20 forecasts, each covering a 48 hours period, running from August 15th through September 10th.

Table 4.15: Errors for forecasts outside Hans using weather forecast from the central site.

No. variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
n/a - Polynomial	253	184	10.2	5.4
1 - Wind Speed	248	182	10.1	5.4
2 - Wind Gust	238	165	9.2	4.0
2 - Wind direction	259	180	10.0	4.0
2 - Humidity	243	174	9.7	4.0
3 - Wind gust and direction	242	168	9.3	3.7
4 - Humidity added	237	162	9.0	2.8

Table 4.16: Errors for forecasts outside Hans using weather forecast from the southern site.

No. variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
n/a - Polynomial	245	171	9.5	4.0
1 - Wind Speed	243	171	9.5	3.9
2 - Wind Gust	234	157	8.7	2.6
2 - Wind direction	253	176	9.5	2.4
2 - Humidity	239	164	9.1	2.5
3 - Wind gust and direction	238	160	8.9	2.3
4 - Humidity added	235	155	8.6	1.5

It is notable how much lower the error was in all cases compared to the Hans storm. This is indicative of the underlying uncertainty of the meteorological prognoses which are input into the models, and shows how much larger the influence of this uncertainty is compared to any improvements offered by models constructed with the random forest algorithm. For example, the lowest NMAE during Hans was 12.7 %, while the highest under more clement conditions was 10.2 %. Nonetheless, there does emerge a pattern of improvement that, while small, mostly matches the expectations set by the historical

data. The exception being wind direction, which likely on account of the forecast being eight-directional did not perform well.

In comparing the outcomes for the Hans even and the normal weeks above, it is clear that there exists a systemic error from the meteorological prognoses. At the same time, it would be reasonable to expect the random forest model to improve outcomes in relative terms - something that is observed for the regular weeks - yet this does not seem to be occuring with any level of reliability in the case of the Hans storm. It is therefore plausible that the model itself is performing worse. However, a larger sample size covering more such events would aid in drawing that conclusion.

Figures 4.10-4.13 show the forecasts for the same models as in the previous section, i.e. polynomial, bivariate, trivariate, and quadrivariate for all forecasts collected from August 15th through August 28th.

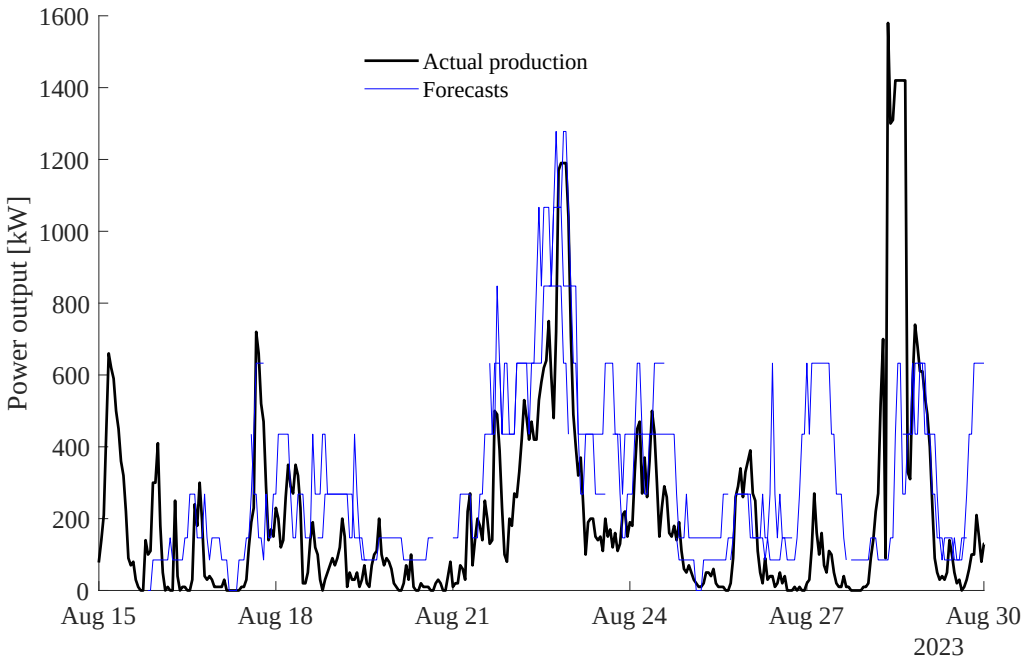

Figure 4.10: Forecasts using the polynomial model compared to actual production.

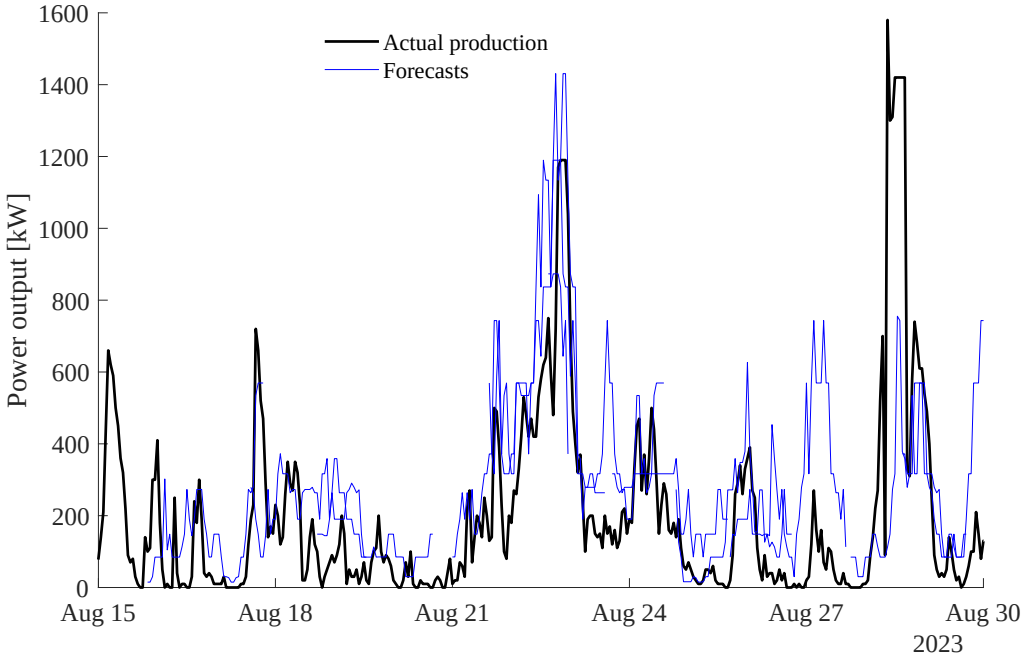

Figure 4.11: Forecasts using the bivariate model with gust winds compared to actual production.

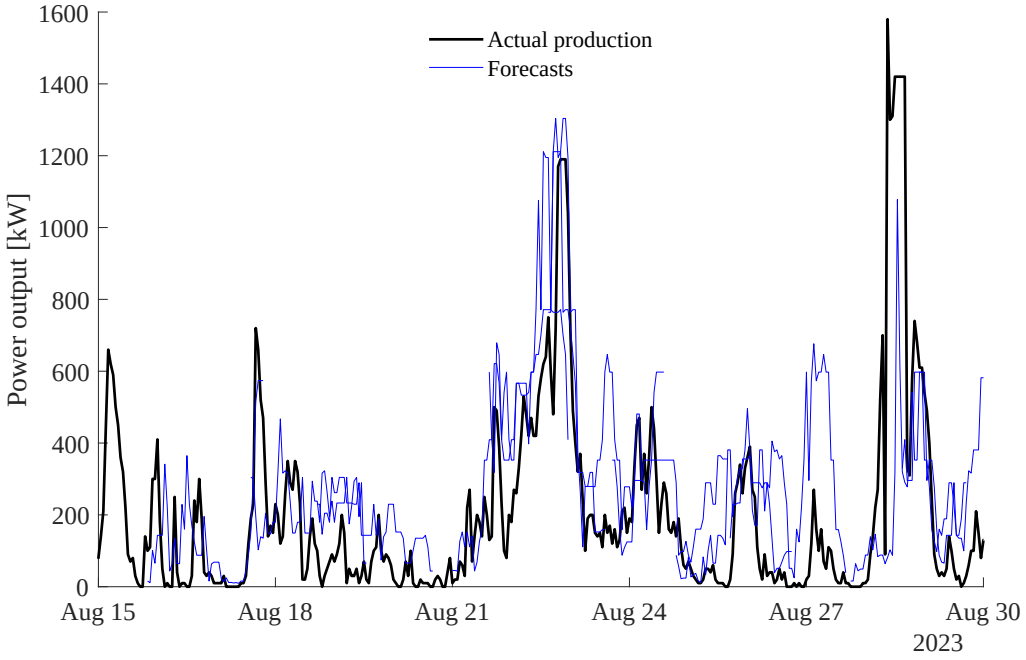

Figure 4.12: Forecasts using the trivaraite compared to actual production.

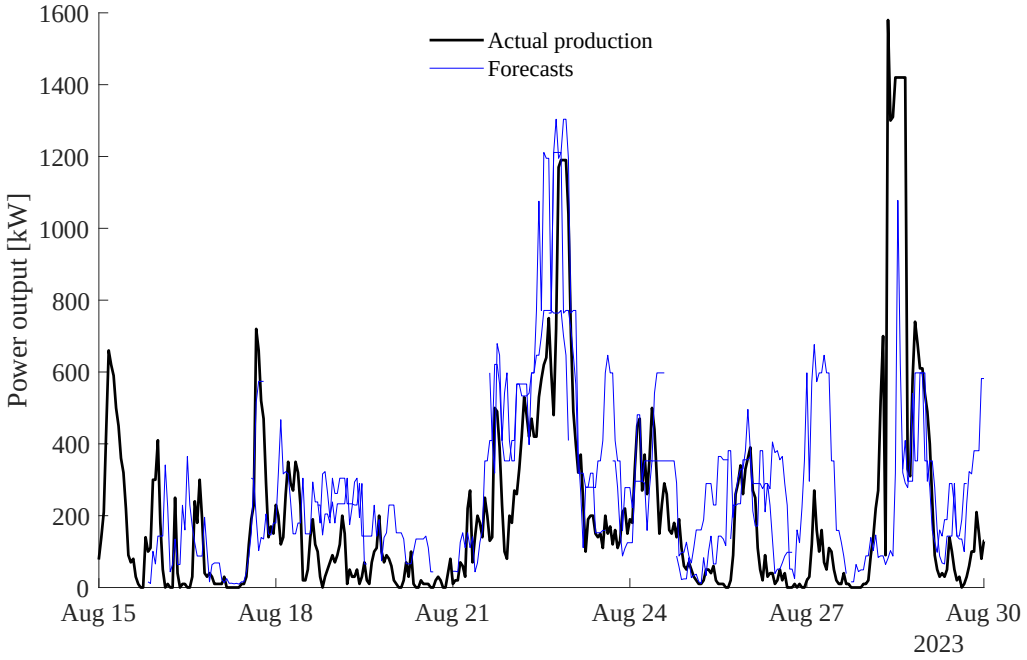

Figure 4.13: Forecasts using the quadrivariate model compared to actual production.

Comparing the above plots does not show an immediately apparent improvement. It is only when the error is calculated over a large timeframe that improvements can be seen. Some interesting observations can be made in relation to the largest errors. For example, consider the very large peak in production on August 28th. Here, the forecasts consistently underestimated production. However, the error is reduced by the multivariate models. For the polynomial model, the highest value was 633 kW. This increased to 755 kW for the bivariate model, 1078 kW for the trivariate model, and 1033 kW for the quadrivariate model. Given that peak production at this time was close to 1600 kW, it is notable that the models taking more variables into account were able to reduce the error.

This is not always the case however. All models overestimated production on August 29th. While actual production was under 200 kW, the polynomial model showed 633 kW, the bivariate a much worse prediction of 743 kW, the trivariate the much better 582 kW, and the quadrivariate 656 kW. At the same time, the multivariate models reached their peaks much later. The polynomial model plateaued at 20:00, yielding its highest value of 655 kW from then on. Yet, at 20:00, the multivariate models yielded a forecast of 569 kW (bivariate), 381 kW (trivariate), and 397 kW (quadrivariate). Thus, the total error over the duration will be reduced somewhat by the multivariate models even though a substantial peak remains.

Similar behavior is exhibited during August 23rd, where all models show sizable peaks in relation to actual production, but the sum of the errors over several hours around 15:00 is reduced, as should be made apparent by the presence of a plateau in the polynomial

model that is not present in the other models. Here, the peaks were 633 kW, 743 kW, 657 kW, and 580 kW.

These are a few of the more notable features, and while not much can be said from this alone, it shows why and in what manner the error was reduced in Tables 4.15 and 4.16 above. In particular, it shows that for any given instance in time, forecasts may be either improved or worsened, and it is only over large timeframes that meaningful improvements materialize.

Figures 4.14-4.17 below show the same as above for forecasts collected on August 29th through September 10th.

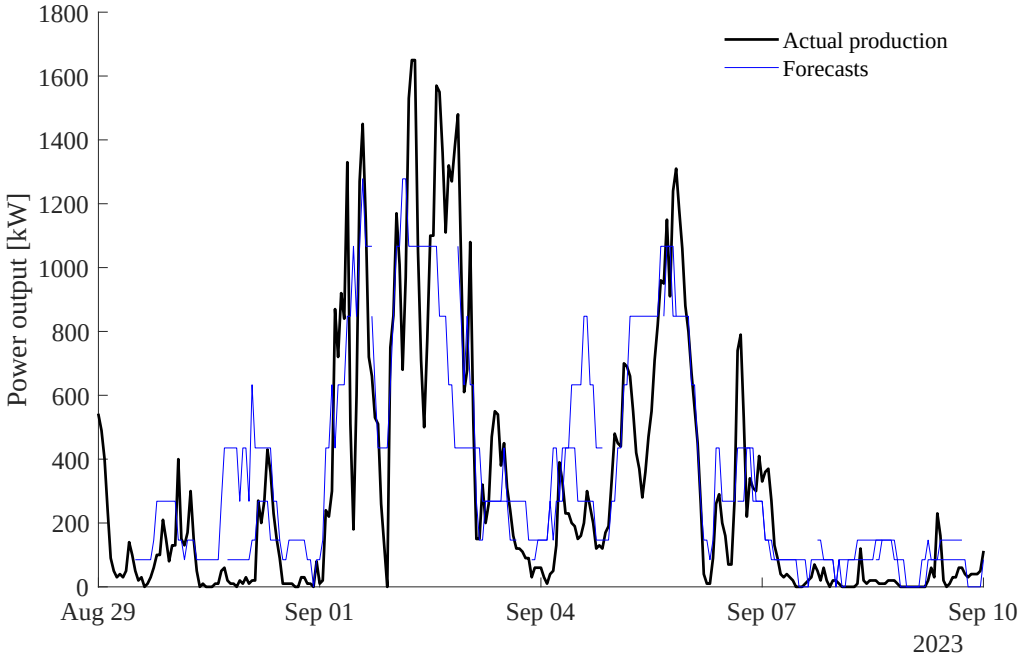

Figure 4.14: Forecasts using the polynomial model compared to actual production.

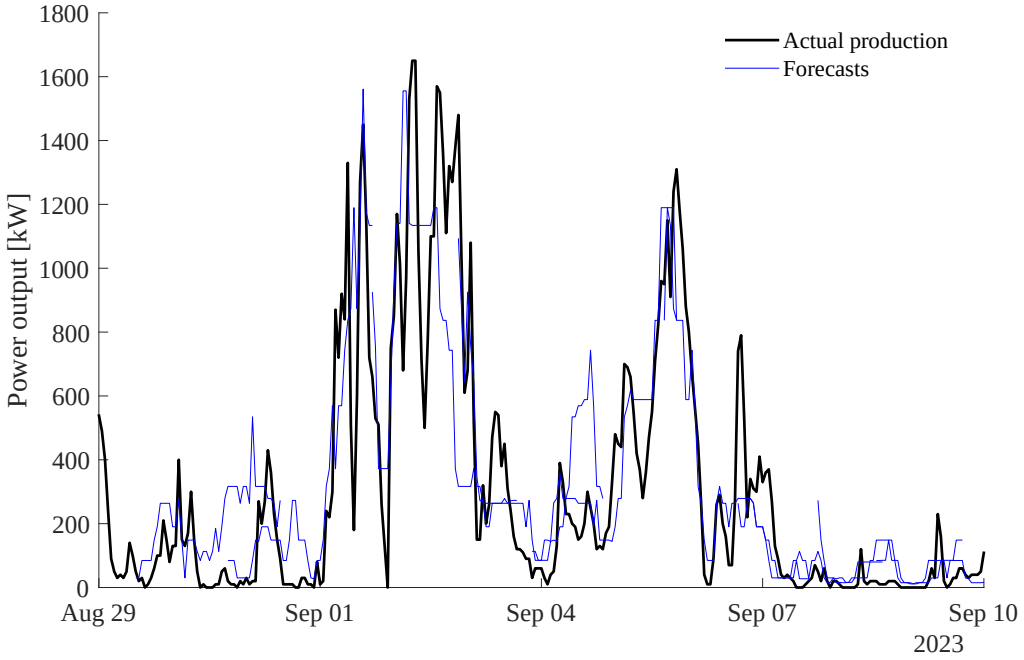

Figure 4.15: Forecasts using the bivariate model with gust winds compared to actual production.

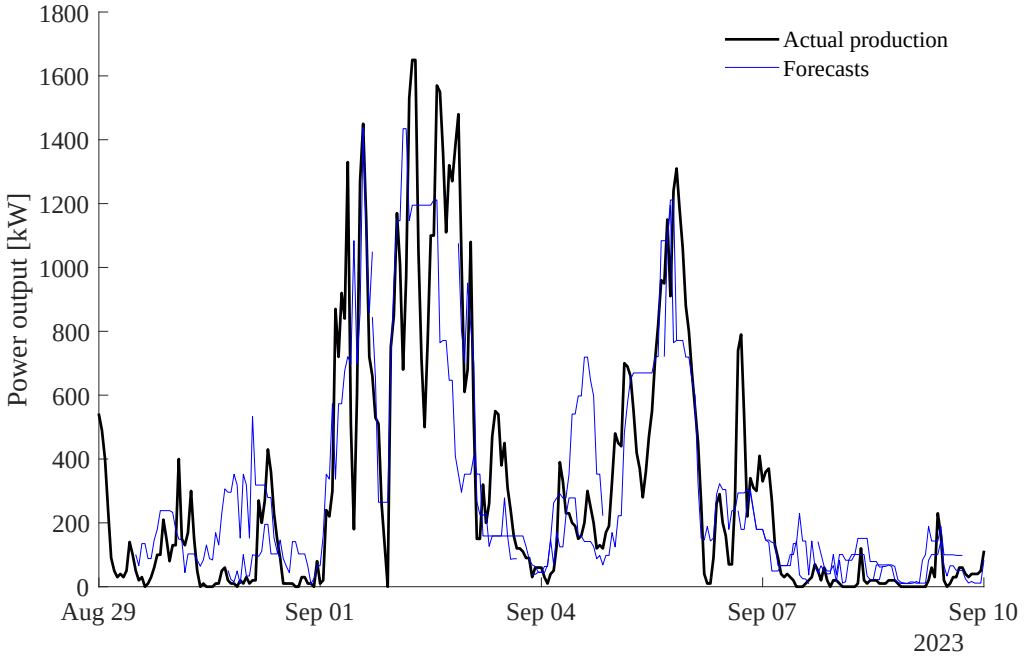

Figure 4.16: Forecasts using the trivaraite compared to actual production.

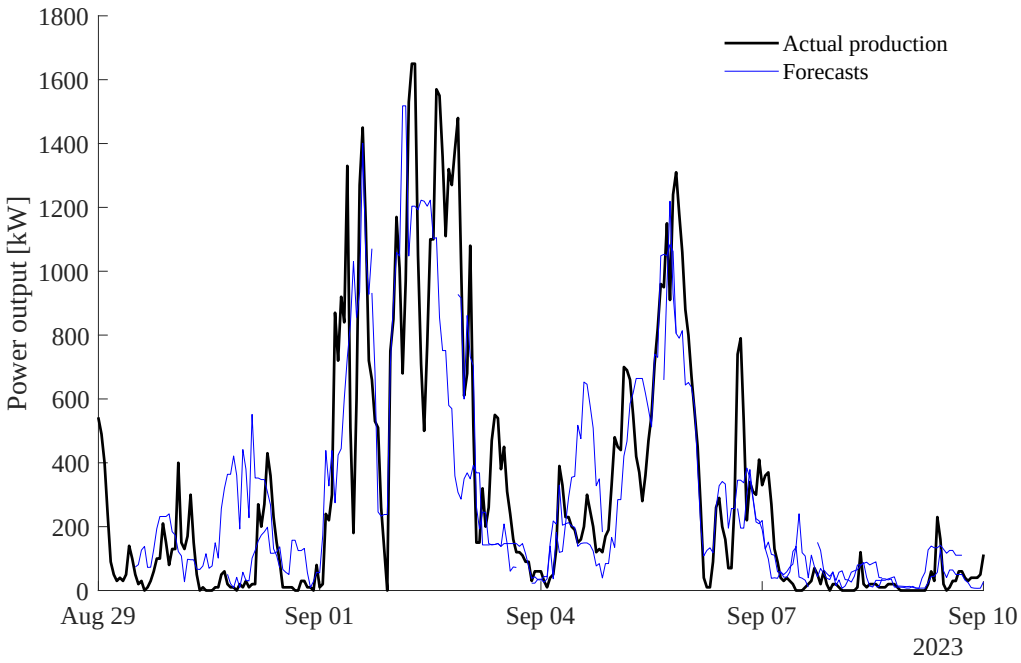

Figure 4.17: Forecasts using the quadrivariate model compared to actual production.

There is a substantial peak over the actual production shown by all four forecasts during September 4th. This peak sits at 848 kW for the polynomial model, 743 kW for the bivariate, 719 kW for the trivariate model, and 646 kW for the quadrivariate model. In this particular case, although the models consistently overestimate production for September 4th, the multivariate models show a decrease in error.

4.9 All weeks

Summing up all the forecasts for all the weeks considered, amounting to a total of 27 forecasts covering a period of time running from August 6th through September 10th in 2023, yields the errors shown in Tables 6.8 and 6.9 below. Overall, wind direction did not provide much of an improvement either as a second variable or as a third, likely on account of the forecasts providing this as an eight-directional value rather than in degrees.

Table 4.17: Errors for all forecasts using weather forecast from the central site.

No. variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
n/a - Polynomial	294	208	11.5	3.5
1 - Wind Speed	295	207	11.5	3.4
2 - Wind Gust	291	197	10.9	2.8
2 - Wind direction	300	207	11.5	2.4
2 - Humidity	291	201	11.2	2.2
3 - Wind gust and direction	284	193	10.8	2.1
4 - Humidity added	278	187	10.4	1.3

Table 4.18: Errors for all forecasts using weather forecast from the southern site.

No. variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
n/a - Polynomial	268	188	10.4	2.8
1 - Wind Speed	270	188	10.5	2.5
2 - Wind Gust	265	180	10.0	1.9
2 - Wind direction	277	190	10.5	1.4
2 - Humidity	265	182	10.1	1.3
3 - Wind gust and direction	262	177	9.9	1.2
4 - Humidity added	262	175	9.7	0.3

Although any improvement seen here is very small, a relatively consistent pattern of small improvements does seem to materialize when considering the various sets of data presented both for historical values and forecasting.

Chapter 5

Conclusion

Considering the outcomes reported above, the random forest algorithm can be concluded to be capable of providing improved predictions for a given dataset for the variables investigated. It is also apparent that the most important factor in producing good wind power forecasts is the dataitself. In this regard, the results show that choosing good data is important. In particular, it appears crucial that much attention is placed on ensuring the highest levels of agreement in meteorological conditions as possible between the following:

1. The turbine site
2. The site from which meteorological data is sourced
3. The site from which meteorological prognoses are sourced

If agreement is poor for any two of these these, it is likely that results will suffer. It is tempting to assume that as long as the latter two agree closely, if it can be assumed that conditions on the turbine site are well correlated (e.g. wind speed has a relatively fixed proportionality given a specific wind direction), a machine learning algorithm can compensate by incorporating the proportionality into itself. In other words, that even if turbine site conditions are different, conditions may still differ in predictable ways. While this may hold to some extent, it does not seem to be enough to overcome the detrimental effects of lack of agreement with site conditions. The influence of different sets of data was greater by far than any impact of using different algorithms, be they a polynomial fit or a machine learning of any number of variables.

This does not mean that machine learning models lack utility in improving wind power forecasting. It simply means that improving the underlying forecasts should take precedence. Finding the data that is the best possible fit for the site should make up most of the work in building a wind power forecasting model, while constructing the model itself is a relatively simple matter.

Unfortunately, the present thesis did not include any data from an array of multiple wind turbines. It can be assumed that such an array would have a much stronger dependence on wind direction, due to wake wind reduction. Accounting for this dependence accurately

would likely require a wind direction forecast in degrees, rather than an eight-directional forecast, which is often provided.

Indeed, the results for the present thesis were not very good when it came to forecasting using wind direction, likely in part as a result of the aforementioned eight-directional forecasting.

An additional point of interest is the apparent difficulty for the multivariate machine learning algorithms in accurately predicting maximum output. This is exemplified in Figures 4.6-4.9 in section 6.3 above, and could also be seen more broadly in predictions based on the historical data that are not included in the report. A possible countermeasure could be to construct a predictive algorithm that relies on machine learning most of the time, but includes an override under certain conditions (such as very high wind speed). More work is required to ascertain whether this would further improve results.

In sum, the present thesis has shown that under certain conditions, the random forest algorithm used on a suitable quantity of historical data - at minimum roughly 15 weeks - is capable of producing predictive models that are capable of outperforming a polynomial regression with cut-offs using the same historical data. These findings are corroborated with actual meteorological prognoses as well, although performance decreased on account of the additional uncertainty introduced. There are indications that further developments taking into account the seasonality of historical data as well as overrides that force the maximum or minimum production values under certain conditions may improve performance further, but more work is needed to verify this. In particular, for seasonality, two full years worth of historical data would be useful, as it would allow for splitting the full year into multiple overlapping sections and test each in sequence to determine the optimal seasonal divides. However, overall, it is the quality of historical data and forecasts that are the most important factor for producing reliable wind power forecasting.

Bibliography

[1] Energimyndigheten, *Energiläget 2015*, https://www.energimyndigheten.se/contentassets/50a0c7046ce54aa88e0151796950ba0a/energilaget-2015_webb.pdf, [Online; accessed 2023-08-23], 2015.

[2] Energimyndigheten, *Energiläget 2022 – en översikt*, https://energimyndigheten.a-w2m.se/Home.mvc?ResourceId=208636, [Online; accessed 2023-08-11], 2022.

[3] European Council, *Infographic - how is eu electricity produced and sold?*, [Online; accessed 2023-08-11]. [Online]. Available: https://www.consilium.europa.eu/en/infographics/how-is-eu-electricity-produced-and-sold/.

[4] C. Anderson, *Wind Turbines - Theory and Practice*. Cambridge University Press, 2020, ISBN: 978-1-108-47832-8.

[5] Vattenfall, *Så fungerar handeln på elbörsen*, https://energyplaza.vattenfall.se/blogg/sa-fungerar-handeln-pa-elborsen, [Online; accessed 2023-08-11].

[6] Energimarknadsinspektionen, *Så här fungerar elmarknaden*, https://ei.se/konsument/el/sa-har-fungerar-elmarknaden, [Online; accessed 2023-08-11].

[7] G. Giebel and G. Kariniotakis, "Wind power forecasting - a review of the state of the art", 2017.

[8] M. Dione and E. Matzner-Löber, "Short-term forecast of wind turbine production with machine learning methods: Direct and indirect approach", 2021.

[9] J. Collins, J. Parkes and A. Tindal, "Forecasting for utility-scale wind farms — the power model challenge", 2009.

[10] SMHI, *Vindhastighet under året och dygnet*, https://www.smhi.se/kunskapsbanken/meteorologi/vind/vindhastighet-under-aret-och-dygnet-1.170858, [Online; accessed 2023-06-30], 2022.

[11] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning Data Mining, Inference, and Prediction, Second Edition*. University of California, 2009.

[12] P. Bickel, F. Goetze and W. Zwet, "Resampling fewer than n observations: Gains, losses, and remedies for losses", *Statistica Sinica*, vol. 7, May 1999. DOI: 10.1007/978-1-4614-1314-1_17.

[13] L. Breiman, "Random forests", 2001.

[14] H. Hansson, "On the evaluation of district heating load predictions", 2023.

[15] LVGMC, *Data search*, https://www.meteo.lv/en/meteorologija-datu-meklesana/?nid=924, [Online; accessed 2023-06-30 through 2023-07-20].

[16] SVT, *Ovädret hans framfart över landet*, https://www.svt.se/nyheter/inrikes/ovadret-hans-framfart-over-landet, [Online; accessed 2023-08-17], 2023.

[17] LVGMC, *Klimata portāls/climate portal*, https://klimats.meteo.lv/pasvaldibu_apskati/valstspilseta/liepajas_pilseta/, [Online; accessed 2023-10-16].

Chapter 6

Appendix

What follows is the raw output from the script used to calculate the various measures for the error that were given in the results. This includes the 'sine extremis' values that were excluded from the results.

6.1 Historical data

Tables 6.1 and 6.2 show the raw output for the random forest models trained on historical data for the central site and the beach site. Table 6.1 correspond to Tables 4.1, 4.3, and 4.5. Table 6.2 corresponds to Tables 4.2, 4.4, and 4.6.

Table 6.1: Original data for statistical measures of the error for models trained on data for the beach site.

No. variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
n/a - Physical model	343.0440	226.5231	12.5151	-12.1229
SE	370.2350	263.1405	14.5382	-14.0884
n/a - Polynomial	219.5316	153.7648	8.4953	0.2492
SE	231.4563	170.5014	9.4200	0.3011
1 - Wind speed only	220.6267	156.3438	8.6378	0.9264
SE	222.7372	159.1844	8.7947	0.9377
1 - Wind speed, adjusted	220.6933	156.5130	8.6952	0.9223
SE	223.0420	159.6707	8.8706	0.9385
1 - Physical model output	221.2700	158.1518	8.7862	1.0221
SE	221.5482	158.5189	8.8066	1.0265
2 - Wind direction	220.6919	143.0347	7.9464	-0.0711
SE	225.7119	149.1651	8.2869	-0.0864
2 - Humidity	221.4486	159.0526	8.8363	1.0419
SE	221.5414	159.1799	8.8433	1.0431
2 - Wind gusts	203.6037	138.2922	7.6829	-0.0291
SE	209.1479	145.3252	8.0736	-0.0566
2 - Wind minimum	220.6592	143.6087	7.9783	-0.0319
SE	224.7930	148.6591	8.2588	-0.0443
2 - Air temperature	221.8981	158.2314	8.7906	1.2618
SE	224.0195	161.0821	8.9490	1.2800
1 - Wind speed only	220.6267	156.3438	8.6378	0.9264
SE	222.7372	159.1844	8.7947	0.9377
2 - Wind gusts added	203.3097	138.0109	7.6249	-0.0342
SE	208.1135	144.1126	7.9620	-0.0621
3 - Wind direction added	202.0939	129.8085	7.1717	-0.4366
SE	208.3863	137.4282	7.5927	-0.4923
4 - Wind minimum added	194.3716	125.9390	6.9580	-0.3235
SE	201.4612	134.6881	7.4413	-0.3704
5 - Air temperature added	184.3851	124.4963	6.8782	-0.1959
SE	190.4676	132.3123	7.3101	-0.2346
5 - Humidity added	183.2951	118.6338	6.5544	-0.4666
SE	189.4218	126.0853	6.9660	-0.5281
6 - All variables	179.1002	120.6841	6.6676	-0.2862
SE	183.8964	126.7399	7.0022	-0.3275

Table 6.2: Original data for statistical measures of the error for models trained on data for the beach site.

No. variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
n/a - Physical model	555.6498	358.8827	19.9379	-19.9263
SE	601.6790	419.9837	23.3324	-23.3223
n/a - Polynomial	283.6042	204.8191	11.3160	0.5657
SE	302.8629	232.9541	12.8704	0.6805
1 - Wind speed only	276.7796	203.1059	11.2213	2.2373
SE	278.5848	205.6140	11.3599	2.2593
1 - Wind speed, adjusted	276.8859	203.3167	11.2954	2.2779
SE	278.6917	205.8267	11.4348	2.3007
1 - Physical model output	278.0646	207.6735	11.5374	2.5257
SE	278.3979	208.1504	11.5639	2.5329
2 - Wind direction	243.5336	173.6099	9.6450	1.9001
SE	244.5877	175.0059	9.7225	1.9168
2 - Humidity	279.1713	200.7065	11.1504	1.8761
SE	279.1713	200.7065	11.1504	1.8761
2 - Wind gusts	246.7132	179.3496	9.9639	2.0368
SE	249.1674	182.7065	10.1504	2.0664
2 - Wind minimum	244.7202	173.6784	9.6488	1.8889
SE	246.1948	175.6199	9.7567	1.9090
2 - Air temperature	279.8602	203.3965	11.2998	2.4308
SE	280.5323	204.3277	11.3515	2.4398
1 - Wind speed only	276.7796	203.1059	11.2213	2.2373
SE	278.5848	205.6140	11.3599	2.2593
2 - Wind gusts added	245.0717	178.6878	9.8723	2.0850
SE	247.9336	182.6276	10.0899	2.1197
3 - Wind direction added	210.4754	150.8475	8.3341	1.6200
SE	212.1532	153.0675	8.4568	1.6375
4 - Wind minimum added	207.5199	146.9809	8.1205	1.4634
SE	209.8896	150.1094	8.2933	1.4839
5 - Air temperature added	204.4421	145.7013	8.0498	1.5377
SE	205.8717	147.5770	8.1534	1.5505
5 - Humidity added	197.2306	139.1848	7.6898	1.1573
SE	198.9945	141.4549	7.8152	1.1648
6 - All variables	196.8749	141.4950	7.8174	1.5342
SE	197.8691	142.8109	7.8901	1.5430

6.2 Seasonality

Table 6.3 shows the full set of data obtained for error measures when seasonality was investigated. It corresponds to Tables 4.8-4.11.

Table 6.3: Statistical measures of the error for models trained on different datasets separated by month for the beach site.

No. variables - Description	Data grouping	MAE (kW)	NMAE (%)	NMBE (%)
	Full year	156.3438	8.6378	0.9264
1 - Wind speed only	March-June	150.7169	8.3269	-0.0611
	July-October	156.1351	8.6263	-0.8323
	September-December	163.1486	9.0137	1.3584
	November-February	181.4685	10.0259	4.3303
	Full year	138.0109	7.6249	-0.0342
2 - Wind gusts added	March-June	134.4377	7.4275	-0.4273
	July-October	137.5541	7.5997	-1.8419
	September-December	141.5954	7.8230	-0.3590
	November-February	158.5975	8.7623	2.9865
	Full year	129.8085	7.1717	-0.4366
3 - Wind direction added	March-June	127.5083	7.0447	-0.3865
	July-October	135.7369	7.4993	-2.5266
	September-December	140.3729	7.7554	-1.5343
	November-February	159.9376	8.8363	2.7845
	Full year	125.9390	6.9580	-0.3235
4 - Wind minimum added	March-June	125.0928	6.9112	-0.3839
	July-October	134.5663	7.4346	-2.3000
	September-December	136.8359	7.5600	-1.4436
	November-February	150.6665	8.3241	2.5314
	Full year	118.6338	6.5544	-0.4666
5 - Humidity added	March-June	118.9481	6.5717	-0.4868
	July-October	131.0246	7.2389	-2.3385
	September-December	132.1882	7.3032	-0.9409
	November-February	143.0595	7.9038	2.3990
	Full year	120.6841	6.6676	-0.2862
6 - All variables	March-June	121.8405	6.7315	-0.4845
	July-October	138.1296	7.6315	-0.3518
	September-December	137.0151	7.5699	0.0639
	November-February	142.7670	7.8877	3.0991

6.3 Forecast data

Figures 6.4-6.5 show the calculated values for the error during the 'Hans' storm without rounding. They correspond to Tables 4.13 and 4.14 in section . Figures 6.6 and 6.7 are the same but for the set of data outside the Hans storm, corresponding to Tables 4.15 and 4.16 respectively. Lastly, Tables 6.8 and 6.9 are for the full month, corresponding to Tables 4.17 and 4.18

Table 6.4: Error for forecasts during Hans using forecasts from the central site.

No. variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
n/a - Polynomial	388.3246	276.2848	15.3492	-2.0539
SE	397.8715	288.8178	16.0454	-2.0854
1 - Wind Speed	397.9510	279.4614	15.5256	-2.0461
SE	400.3195	282.3747	15.6875	-2.0455
2 - Wind Gust	399.0310	283.7269	15.7626	-0.7015
SE	399.6192	284.4445	15.8025	-0.6964
2 - Wind direction	394.6399	285.1968	15.8443	-2.0143
SE	406.2254	300.5168	16.6954	-2.0381
2 - Humidity	400.4981	281.7084	15.6505	-2.9175
SE	400.4981	281.7084	15.6505	-2.9175
3 - Wind gust and direction	374.6470	261.8607	14.5478	-2.2838
SE	377.4446	265.3667	14.7426	-2.2943
4 - Humidity added	369.1235	257.5839	14.3102	-3.0489
SE	374.1147	263.6860	14.6492	-3.0779

Table 6.5: Errors for forecasts during Hans using forecasts from the southern site.

No. variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
n/a - Polynomial	324.2615	233.9520	12.9973	-0.7826
SE	332.2343	244.7746	13.5986	-0.7731
1 - Wind Speed	334.6406	237.5056	13.1948	-1.5838
SE	334.6406	237.5056	13.1948	-1.5838
2 - Wind Gust	337.9035	246.0820	13.6712	-0.1290
SE	340.4339	249.5067	13.8615	-0.1318
2 - Wind direction	341.0451	244.0059	13.5559	-2.0689
SE	348.3299	253.4968	14.0832	-2.0968
2 - Humidity	324.0375	231.6099	12.8672	-2.0767
SE	324.0375	231.6099	12.8672	-2.0767
3 - Wind gust and direction	317.6552	228.0896	12.6716	-1.7130
SE	321.9614	233.6802	12.9822	-1.7356
4 - Humidity added	327.7182	235.5275	13.0849	-3.1748
SE	332.1784	241.5275	13.4182	-3.2521

Table 6.6: Errors for forecasts outside Hans using weather forecast from the central site.

No. variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
n/a - Polynomial	252.9271	183.9426	10.2190	5.4446
SE	255.4676	187.6460	10.4248	5.5551
1 - Wind Speed	248.3055	181.9938	10.1108	5.3553
SE	248.3055	181.9938	10.1108	5.3553
2 - Wind Gust	238.4700	165.4436	9.1913	3.9482
SE	240.2210	167.6668	9.3148	3.9941
2 - Wind direction	258.5279	179.8822	9.9935	3.9920
SE	261.3938	183.5993	10.2000	4.0642
2 - Humidity	242.8224	174.1816	9.6768	3.9731
SE	242.8224	174.1816	9.6768	3.9731
3 - Wind gust and direction	242.1690	167.9583	9.3310	3.7307
SE	245.1167	171.8100	9.5450	3.8072
4 - Humidity added	236.7631	161.6951	8.9831	2.8232
SE	240.2854	166.2402	9.2356	2.8909

Table 6.7: Errors for forecasts outside Hans using weather forecast from the southern site.

No. variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
n/a - Polynomial	244.6710	171.4479	9.5249	4.0353
SE	248.0524	176.1884	9.7882	4.1494
1 - Wind Speed	242.9783	171.3447	9.5191	3.9188
SE	242.9783	171.3447	9.5191	3.9188
2 - Wind Gust	234.1936	156.7199	8.7067	2.6142
SE	237.6729	161.0164	8.9454	2.6698
2 - Wind direction	253.0258	171.6041	9.5336	2.3782
SE	255.5611	174.8309	9.7128	2.4130
2 - Humidity	239.4808	164.2499	9.1250	2.5485
SE	239.4808	164.2499	9.1250	2.5485
3 - Wind gust and direction	238.4364	159.7178	8.8732	2.2573
SE	242.5038	164.8674	9.1593	2.3153
4 - Humidity added	234.6228	154.8339	8.6019	1.4725
SE	238.6283	159.9127	8.8840	1.5096

Table 6.8: Errors for all forecasts using weather forecast from the central site.

No. variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
n/a - Polynomial	294.0782	207.8832	11.5491	3.5005
SE	298.1168	213.3201	11.8511	3.6162
1 - Wind Speed	294.5183	207.1427	11.5079	3.4153
SE	294.9687	207.6878	11.5382	3.4312
2 - Wind Gust	290.5230	196.5121	10.9173	2.7588
SE	292.3261	198.7572	11.0421	2.7852
2 - Wind direction	300.0099	207.1478	11.5082	2.4211
SE	304.7132	213.0037	11.8335	2.5140
2 - Humidity	291.0608	201.1780	11.1766	2.1508
SE	291.2830	201.4408	11.1912	2.1568
3 - Wind gust and direction	283.9468	192.9018	10.7168	2.1366
SE	286.9425	196.7195	10.9289	2.1753
4 - Humidity added	278.4526	187.1235	10.3957	1.3370
SE	282.9372	192.6711	10.7040	1.3861

Table 6.9: Errors for all forecasts using weather forecast from the southern site

No. variables - Description	RMSE (kW)	MAE (kW)	NMAE (%)	NMBE (%)
n/a - Polynomial	267.5884	187.6527	10.4251	2.7862
SE	272.0214	193.6905	10.7606	2.8933
1 - Wind Speed	269.7810	188.4780	10.4710	2.4704
SE	269.7810	188.4780	10.4710	2.4704
2 - Wind Gust	265.1310	179.5490	9.9749	1.8636
SE	268.5603	183.8562	10.2142	1.8946
2 - Wind direction	277.3372	189.7652	10.5425	1.3543
SE	280.6886	193.9007	10.7723	1.3964
2 - Humidity	264.6803	181.7101	10.0950	1.2769
SE	264.6803	181.7101	10.0950	1.2769
3 - Wind gust and direction	261.5550	177.4679	9.8593	1.2363
SE	265.4656	182.4797	10.1378	1.2607
4 - Humidity added	261.8843	175.2074	9.7337	0.3159
SE	266.2185	180.6972	10.0387	0.3235