

# Data-driven multivariate wind turbine performance modeling

Refining wind turbine performance estimations for atmospheric conditions by using machine learning

D.L. van der Arend

Master of Science Thesis





# **Data-driven multivariate wind turbine performance modeling**

**Refining wind turbine performance estimations for atmospheric  
conditions by using machine learning**

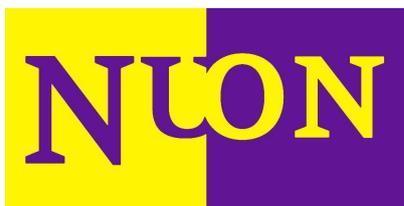
MASTER OF SCIENCE THESIS

For obtaining the degree of Master of Science in Aerospace Engineering  
at Delft University of Technology

D.L. van der Arend

March 2, 2018

Wind Energy Research Group · Faculty of Aerospace Engineering · Delft University of  
Technology



**VATTENFALL**



The work in this thesis was supported by NUON, part of Vattenfall. Their cooperation is hereby gratefully acknowledged.



Copyright © Aerodynamics & Wind Energy (AWE)  
All rights reserved.

DELFT UNIVERSITY OF TECHNOLOGY  
DEPARTMENT OF  
AERODYNAMICS & WIND ENERGY (AWE)

The undersigned hereby certify that they have read and recommend to the Faculty of  
Aerospace Engineering for acceptance a thesis entitled

DATA-DRIVEN MULTIVARIATE WIND TURBINE PERFORMANCE MODELING

by

D.L. VAN DER AREND

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN AEROSPACE ENGINEERING AT DELFT UNIVERSITY OF  
TECHNOLOGY

Dated: March 2, 2018

Chairman of committee:

\_\_\_\_\_  
Prof.dr. S.J. Watson

Supervisor:

\_\_\_\_\_  
Dr.ir. W.A.A.M. Bierbooms

Reader:

\_\_\_\_\_  
Dr.Ing. J. Kober

Reader:

\_\_\_\_\_  
Drs. J.P. Coelingh



---

# Summary

As the interest in wind energy increases, more (potential) wind farm sites are put up for tender. With more actors in the market participating in these tenders and subsidy schemes changing, accurate prediction of the (life-time) energy yield for new wind farms is of increased importance as this estimation is the base of business cases for these tenders. However, to be able to set up more accurate predictions of the energy yield of a wind turbine or farm, the influence of atmospheric parameters on turbine performance and therefore energy yield needs better understanding. Taking into account these additional (external) factors would refine the prediction to the site conditions, yielding in more accurate initial energy yield predictions.

Since theoretical knowledge of these influences is not always sufficiently present in literature or, if even present, not always agreed upon in consensus, another approach is taken to obtain more insight in the impact of atmospheric parameters on the turbine performance. By analyzing empirical data from existing wind farms and combining this with machine learning methodology, empirical dependencies are obtained yielding in multivariate power curve modeling of a wind turbine. In recent work Pelletier et al. [1] and Janssens et al. [2] have researched such multivariate power curve modeling through machine learning with success. This thesis goes one step further in this non-parametric approach and looks into feasibility of using the constructed multivariate power curve to assess individual influence of atmospheric parameters on turbine performance, improve power curve verification and refine long-term energy yield estimations for these atmospheric parameters.

This approach results in three different models, which are based on a neural network, which is a type of machine learning algorithm designed to learn from large amounts of data. The first model is solely used for investigation of the influence of singled out atmospheric parameters on wind turbine performance. The second model is an application of the first model and refines the performance prediction of a single wind turbine to improve power curve verification. The third model uses the modeled relations to refine the prediction of the long-term energy yield estimation for new wind farms, by refining the warranted power curve to site conditions.

Despite more empirical data is required -especially a larger variety of different wind farms- to validate and test these models for more wind farms, the method does shown promise in modeling the multivariate power curve accounting for wind speed, turbulence intensity,

wind shear, wind veer, ambient temperature, ambient pressure and relative humidity. Found correlations between these atmospheric parameters and the turbine power are generally in accordance with literature and theory, if available. This multivariate power curve modeling is successfully adapted to refine future turbine performance estimations for individual turbines. Modeling the refinement of the long-term energy yield estimation does show improved results with respect to current methods, but is currently only tested with three wind farms, limiting the generalized application of the model on wind farms dissimilar to the wind farms for which it was trained. Possible application of these models should therefore be performed with care.

---

# Table of Contents

<b>Glossary</b>	<b>xv</b>
List of Acronyms . . . . .	xv
List of Symbols . . . . .	xviii
<b>Preface</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
1-1 Thesis motivation and research goals . . . . .	2
1-2 Thesis outline . . . . .	4
<b>2 Wind turbine performance</b>	<b>5</b>
2-1 Turbine performance prediction . . . . .	5
2-2 Influences on wind turbine performance . . . . .	8
2-3 Power curve modeling using machine learning . . . . .	14
2-4 Discussion on power curve modeling . . . . .	16
<b>3 Machine learning</b>	<b>17</b>
3-1 Method trade-off . . . . .	17
3-2 Artificial neural network . . . . .	19
3-3 Discussion on machine learning method . . . . .	28
<b>4 Wind farm conditions</b>	<b>29</b>
4-1 Site descriptions . . . . .	29
4-2 Site calibration and power curve verification . . . . .	32
4-3 Power curve verification . . . . .	32
4-4 Measurement sectors . . . . .	33
4-5 Discussion on measurement conditions . . . . .	36

<b>5</b>	<b>Pre-processing</b>	<b>37</b>
5-1	Data filtering . . . . .	37
5-2	Atmospheric influences . . . . .	41
5-3	Filtered dataset evaluation . . . . .	42
5-4	(Filtered) measured performance . . . . .	43
5-5	Network input and output . . . . .	44
5-6	Discussion on data pre-processing . . . . .	46
<b>6</b>	<b>Setting up a neural network based model</b>	<b>47</b>
6-1	Neural network software . . . . .	47
6-2	Model overview . . . . .	48
6-3	Hyper-parameters . . . . .	49
6-4	Discussion on optimized models . . . . .	55
<b>7</b>	<b>Model results and applicability</b>	<b>57</b>
7-1	Power curve verification model . . . . .	57
7-2	Performance Correlation Assessment model . . . . .	61
7-3	Business Case Refinement model . . . . .	68
7-4	Limitations and shortcomings . . . . .	74
<b>8</b>	<b>Conclusions and recommendations</b>	<b>75</b>
8-1	Conclusions . . . . .	75
8-2	Recommendations . . . . .	77
<b>A</b>	<b>Wind farm Prinses Alexia</b>	<b>79</b>
A-1	Wind farm site . . . . .	79
A-2	Measurement equipment . . . . .	81
A-3	Power curve verification . . . . .	82
A-4	Measurement sectors . . . . .	82
A-5	Pre-processing . . . . .	84
<b>B</b>	<b>Wind farm Pen y Cymoedd</b>	<b>85</b>
B-1	Wind farm site . . . . .	85
B-2	Measurement equipment . . . . .	87
B-3	Site calibration and power curve verification . . . . .	89
B-4	Measurement sectors . . . . .	90
B-5	Pre-processing . . . . .	93
<b>C</b>	<b>Wind farm Norrekær Enge I</b>	<b>95</b>
C-1	Wind farm site . . . . .	95
C-2	Measurement equipment . . . . .	96
C-3	Power curve verification . . . . .	96
C-4	Measurement sectors . . . . .	97
C-5	Pre-processing . . . . .	100

---

<b>D</b>	<b>Considered machine learning methods</b>	<b>101</b>
D-1	Support vector regression . . . . .	101
D-2	Fuzzy logic . . . . .	105
D-3	Regression/decision trees . . . . .	108
<b>E</b>	<b>Neural network function influences</b>	<b>111</b>
E-1	Optimization function . . . . .	111
E-2	Activation function . . . . .	113
E-3	Normalization scheme . . . . .	116
<b>F</b>	<b>Grid search optimizations</b>	<b>121</b>
F-1	Epoch and batch size . . . . .	122
F-2	Early stopping epochs . . . . .	123
<b>G</b>	<b>Numerical sensitivity and correlation on measurements</b>	<b>125</b>
G-1	Correlation coefficient . . . . .	125
G-2	Numerical sensitivity versus correlation comparison . . . . .	126
<b>H</b>	<b>Singled out parameter impact on power curve</b>	<b>133</b>
	<b>Bibliography</b>	<b>139</b>



---

# List of Figures

1-1	Global wind energy market forecast up to 2021 via Global Wind Energy Council [3]	1
2-1	Wind turbine's power curve on the left, measured turbine performance on the right [4]	7
2-2	Schematic representation of the inner-outer range proposal of the Power curve working group (PCWG) [5]	8
2-3	Example of measured influences on variables for 3-4 m/s bin. [1]	8
2-4	General effect of turbulence intensity on wind turbine performance. [6]	8
2-5	Ekman spiral visualized (via <a href="http://techniklexikon.net">techniklexikon.net</a> )	10
2-6	Percentage of change in power production (scale on the right) for a range of wind speed and direction shear, using hub height wind speed of 8 m/s (top) and 10 m/s (bottom). Power change interval is 0.5%, the thick black line represents a zero change contour. [7]	10
2-7	Blade element and its perceived wind speed with resulting forces (via lectures Michel Zaaier at Delft University of Technology (TU Delft)).	11
2-8	Comparison of power production for a yawed and tilted. Note that yaw angle $\varphi$ is different from blade element inflow angle $\varphi$ . [8]	11
3-1	Inside a neural network node (via <a href="http://www.datacamp.com">www.datacamp.com</a> )	19
3-2	Simple feed-forward (artificial) neural network architecture (network topology) with a single hidden layer [9]	20
3-3	Fluctuations in training using Stochastic gradient descent (SGD) (via [10])	24
3-4	Top: classical momentum method. Bottom: Nesterov accelerated momentum [11]	24
3-5	Batch size and learning rate impact on accuracy for image recognition [12]	25
3-6	Linear (striped line) and n-th order polynomial (uninterrupted line) regression (via KDnuggets.com, modified by author)	26
3-7	Typical learning (lower curve) and testing/validation (upper curve) errors. [13]	26
4-1	Locations of Pen y Cymoedd (PyC), Prinses Alexia and Norrekær Enge I (NKE).	30

4-2	International Electrotechnical Commission (IEC) standard on location and valid measurement sector for site calibration [14]	34
4-3	Wind sectors for turbine R01, in Prinses Alexia, and paired met mast. <sup>2</sup>	35
4-4	Wind sectors for turbine R02, in Prinses Alexia, and paired met mast. <sup>2</sup>	35
4-5	Wind sectors for turbine T82, in PyC, and paired met mast (at the center of the wind rose). <sup>2</sup>	35
4-6	Wind sectors for turbine T83, in PyC, and paired met mast (at the center of the wind rose). <sup>2</sup>	35
4-7	Wind sectors for turbines T04 and T05, in NKE, and paired meteorological mast (at the center of wind rose).	35
5-1	Measured performance in the free stream sector and warranted power curve for Prinses Alexia	44
5-2	Measured performance in the free stream sector and warranted power curve for PyC	44
5-3	Measured performance in the free stream sector and warranted power curve for NKE	44
6-1	Flowchart of all intended models, each indicated in green.	48
6-2	5-fold cross validation. The dataset remains the same for each training process, but the training proces itself is evaluated 5 times for each setting, averaging the results per setting.	51
6-3	Batch size grid search test dataset results, evaluated per farm individually.	53
6-4	Early stopping epoch amount grid search test dataset results, evaluated per farm individually.	54
6-5	Dropout rate grid search test dataset results, evaluated per farm individually.	54
6-6	Weight penalty search test dataset results, evaluated per farm individually.	55
7-1	Power curve verification (PCV) model training error/loss function history.	58
7-2	Performance deviation of measured power from warranted power for Prinses Alexia test subset.	59
7-3	PCV model performance deviation of modeled power from warranted power for Prinses Alexia test subset.	59
7-4	Histogram of predicted power using Warranted power curve (WPC) minus measured power for Prinses Alexia test set. Plotted with 75 bins.	60
7-5	Histogram of predicted power using modeled power minus measured power for Prinses Alexia test set. Plotted with 75 bins. Modeld with PCV model.	60
7-6	Change of PCA model accuracy with one parameter removed from the input parameters. Accuracy shown in (a) Root mean squared error (RMSE) and (b) Mean absolute error (MAE)	62
7-7	Numerical sensitivity along power curve, singled out for ambient temperature. The neural model is trained for Prinses Alexia.	65
7-8	Numerical sensitivity along power curve, singled out for Turbulence intensity (TI). The neural model is trained for Prinses Alexia.	65
7-9	Numerical sensitivity along power curve, singled out for wind shear component. The neural model is trained for Prinses Alexia.	66
7-10	Numerical sensitivity along power curve, singled out for wind veer. The neural model is trained for Prinses Alexia.	67

7-11	Numerical sensitivity along power curve, singled out for humidity. The neural model is trained for Prinses Alexia. . . . .	67
7-12	Numerical sensitivity along power curve, singled out for ambient pressure. The neural model is trained for Prinses Alexia. . . . .	68
7-13	BCR model training error/loss function history. . . . .	69
7-14	Difference between measured and warranted power for Prinses Alexia vs. wind speed. . . . .	70
7-15	Difference between modeled and warranted power for Prinses Alexia vs. wind speed, modeled with BCR model. . . . .	70
7-16	Difference between measured and warranted power for PyC vs. wind speed. . . . .	70
7-17	Difference between modeled and warranted power for PyC vs. wind speed, modeled with BCR model. . . . .	70
7-18	Difference between measured and warranted power for NKE vs. wind speed. . . . .	70
7-19	Difference between modeled and warranted power for NKE vs. wind speed, modeled with BCR model. . . . .	70
7-20	Histogram of error between warranted power and measured power for Prinses Alexia test set. Plotted with 75 bins. . . . .	71
7-21	Histogram of error between modeled power and measured power for Prinses Alexia test set. Plotted with 75 bins, modeled with BCR model. . . . .	71
7-22	Histogram of error between warranted power and measured power for PyC test set. Plotted with 75 bins. . . . .	71
7-23	Histogram of error between modeled power and measured power for PyC test set. Plotted with 75 bins, modeled with BCR model. . . . .	71
7-24	Histogram of error between warranted power and measured power for NKE test set. Plotted with 75 bins. . . . .	72
7-25	Histogram of error between modeled power and measured power for NKE test set. Plotted with 75 bins, modeled with BCR model. . . . .	72
A-1	Location of wind farm Prinses Alexia in The Netherlands. Map via Google Maps. . . . .	80
A-2	Prinses Alexia location of met mast and relevant turbines. (from [15], edited by author) . . . . .	81
A-3	Wind sectors for turbine R01 and paired meteorological mast. <sup>1</sup> . . . . .	83
A-4	Wind sectors for turbine R02 and paired meteorological mast. <sup>1</sup> . . . . .	83
B-1	Location of PyC. Map via Google Maps. . . . .	86
B-2	PyC windfarm location of meteorological masts and turbines [16] . . . . .	88
B-3	Additional wind turbines at the South-East border of PyC, map via Google maps . . . . .	89
B-4	Distance of turbine T83 to met mast paired to turbine T82. <sup>1</sup> . . . . .	91
B-5	Distance of turbine T82 to met mast paired to turbine T83. <sup>1</sup> . . . . .	91
B-6	Wind direction sectors from reference mast paired to T82 <sup>2</sup> . . . . .	91
B-7	Wind direction sectors from reference mast paired to T83 <sup>3</sup> . . . . .	92
C-1	Location of Norrekær Enge I, its turbines and the (later placed) meteorological mast. Maps via Google Maps. . . . .	96
C-2	Location of the later placed met mast in NKE. [17] . . . . .	96

C-3	10-Minute averaged measured temperature at met mast and at the hub (from Supervisory Control And Data Acquisition (SCADA). The red line is the met mast shifted by -60 minutes.) . . . . .	98
C-4	Assessed influence directions of turbines on the meteorological mast. . . . .	98
C-5	Wind direction sectors for turbines T04 and T05, paired to the met mast. Map via Google Maps. . . . .	99
D-1	Linear SVM, the gray area indicates the $\varepsilon$ 'tube' [18] . . . . .	102
D-2	Soft margin loss setting for a linear Support vector machine (SVM), via [18] . . .	103
D-3	Comparison of fitting capabilities to training data of linear, polynomial and Radial basis functions (RBF) kernel functions. (via Python Scikit package examples) . .	106
D-4	Overview of fuzzy interference system [19] . . . . .	106
E-1	Logistic/sigmoid activation function . . . . .	115
E-2	Tanh activation function . . . . .	115
E-3	Rectified linear unit (ReLU) activation function . . . . .	115
E-4	Exponential linear unit (ELU) activation function . . . . .	115
E-5	Gaussian activation function . . . . .	115
E-6	Softsign activation function . . . . .	115
E-7	Difference of measured power from warranted power for Prinses Alexia test set using a normalization to rated power scheme. . . . .	118
E-8	Difference of modeled power from warranted power for Prinses Alexia test set using a normalization to rated power scheme. . . . .	118
E-9	Difference of measured power from warranted power for Prinses Alexia test set using a normalization to difference from WPC scheme. . . . .	118
E-10	Difference of modeled power from warranted power for Prinses Alexia test set using a normalization to difference from WPC scheme. . . . .	118
E-11	Histogram of the difference between modeled and measured power for normalization to rated power, for the test dataset of Prinses Alexia. . . . .	120
E-12	Histogram of the difference between modeled and measured power for normalization to the difference from warranted power, for the test dataset of Prinses Alexia. . . . .	120
F-1	PCV model batch size grid search, evaluated for the test dataset of Prinses Alexia. . . . .	122
F-2	Test dataset performance for varying early stopping epochs for Prinses Alexia. . .	123
G-1	Pearson correlation (a) and numerical sensitivity (b) of individual parameters over the course of the power curve, right axis, for Prinses Alexia. . . . .	127
G-2	Correlation and numerical sensitivity of turbulence intensity along power curve for Prinses Alexia measured data. . . . .	128
G-3	Correlation and numerical sensitivity of wind shear along power curve for Prinses Alexia measured data. . . . .	129
G-4	Correlation and numerical sensitivity of ambient temperature along power curve for Prinses Alexia measured data. . . . .	129
G-5	Correlation and numerical sensitivity of turbulence intensity along power curve for Prinses Alexia measured data. . . . .	130

---

G-6	Correlation and numerical sensitivity of relative humidity along power curve for Prinses Alexia measured data. . . . .	131
G-7	Correlation and numerical sensitivity of wind veer along power curve for Prinses Alexia measured data. . . . .	131
H-1	Individual influence of TI on performance for Prinses Alexia via measurements. . .	134
H-2	Individual influence of TI on performance for Prinses Alexia via model. . . . .	134
H-3	Individual influence of temperature on performance for Prinses Alexia via measurements. . . . .	134
H-4	Individual influence of temperature on performance for Prinses Alexia via model. . .	134
H-5	Individual influence of wind shear on performance for Prinses Alexia via measurements. . . . .	135
H-6	Individual influence of wind shear on performance for Prinses Alexia via model. . .	135
H-7	Individual influence of pressure on performance for Prinses Alexia via measurements. . .	136
H-8	Individual influence of pressure on performance for Prinses Alexia via model. . . . .	136
H-9	Individual influence of humidity on performance for Prinses Alexia via measurements. . .	136
H-10	Individual influence of relative humidity on turbine performance for Prinses Alexia . . .	136
H-11	Individual influence of wind veer on performance for Prinses Alexia via measurements. . .	137
H-12	Individual influence of wind veer on performance for Prinses Alexia via model. . . . .	137



---

# List of Tables

2-1	Model results for univariate and multivariate PC modeling [20]	16
3-1	Multiple criteria analysis trade-off between four selected methods for machine learning regression	18
4-1	Site conditions at hub height for Prinses Alexia, PyC and NKE, respectively from [15], [21] and [22]	30
4-2	Measurement equipment altitudes located in masts linked to turbines, respectively via [15], [23] and NKE dataset.	31
4-3	Turbine specifications	31
4-4	Wind velocity calibration relations [24]	32
4-5	Turbine wind sector ranges	34
5-1	Classification of data points [25]	38
5-2	Applied filters and their order	38
5-3	Prinses Alexia turbines lost data samples due to filtering in free stream	43
5-4	Prinses Alexia turbines lost data samples due to filtering in wake stream	43
6-1	Grid search hyper-parameters initial settings	52
6-2	Model performance after hyper-parameter grid searches. The test set contains the equally sized test sets for all three wind farms combined.	52
6-3	Grid search parameters for regularization parameters	56
7-1	Performance of PCV model results vs. WPC for Prinses Alexia and PyC test dataset.	60
7-2	PCA model accuracy after leaving out single input parameter, including ranking of parameter importance.	62
7-3	Constant value and range of evaluated parameters, trained using Prinses Alexia dataset.	64
7-4	Relative performance between WPC method and BCR model quantified for Prinses Alexia, PyC and NKE test sets.	73

A-1	Turbine specifications Senvion 3.4M104, via [15]	80
A-2	Turbine specifications Senvion 3.4M104, via [15]	81
A-3	Turbine wind sector ranges	82
A-4	Prinses Alexia turbines lost data-points due to filtering in free stream	84
A-5	Prinses Alexia turbines lost data-points due to filtering in wake stream	84
B-1	Site specific conditions at hub height [21]	86
B-2	PyC turbine specifications. [23]	87
B-3	Measurement equipment altitudes for both types of reference and calibration masts [23]	87
B-4	Wind velocity calibration relations. [24]	89
B-5	Distances and blocked sector between met masts and turbines	90
B-6	Turbine wind sector ranges for PyC.	90
B-7	Lost data-points due to filtering in free stream	93
B-8	Lost data-points due to filtering in wake stream	93
C-1	Siemens SWT-2.3-93 specifications	95
C-2	Measurement equipment and altitude. Data via [17] and NKE datasets	97
C-3	Turbine wind sector ranges for NKE	99
C-4	NKE turbines lost data-points due to filtering in free stream	100
C-5	NKE turbines lost data-points due to filtering in wake stream	100
D-1	Comparison between neural network and fuzzy logic systems [26]	107
E-1	Settings for each run in training function evaluation.	112
E-2	Average computational time and optimizer accuracy, trained for Prinses Alexia and averaged for 10 runs.	112
E-3	<i>Keras</i> activation functions and range [27]	113
E-4	Normalization scheme comparing network settings	117
E-5	Resulting performance of trained model using only data from Prinses Alexia	119
F-1	Grid search hyper-parameters initial settings	121

---

# Glossary

## List of Acronyms

<b>Adagrad</b>	Adaptive Gradient algorithm
<b>Adam</b>	Adaptive moment estimation
<b>AEP</b>	Annual energy production
<b>ANFIS</b>	Adaptive neuro-fuzzy interference system
<b>ANN</b>	Artificial neural network
<b>AoA</b>	Angle of attack
<b>API</b>	Application programming interface
<b>BCR</b>	Business Case Refinement
<b>BEM</b>	Blade element momentum
<b>BGD</b>	Batch gradient descent
<b>CCFL</b>	Cluster center fuzzy logic
<b>CCR</b>	Climatic condition review
<b>CFD</b>	Computational fluid dynamics
<b>DNV-GL</b>	GL Garrad Hassan Deutschland GmbH
<b>ELU</b>	Exponential linear unit
<b>FIS</b>	Fuzzy interference system
<b>IEC</b>	International Electrotechnical Commission
<b>k-NN</b>	k-Nearest neighbor
<b>LCL</b>	Lower control limit

---

<b>LiDAR</b>	Light Detection And Ranging
<b>LLJ</b>	Lower level jet-stream
<b>MAE</b>	Mean absolute error
<b>MAPE</b>	Mean absolute percentage error
<b>MBGD</b>	Multi-batch gradient descent
<b>MF</b>	Membership function
<b>MLP</b>	Multi-layer perceptron
<b>MoB</b>	Method of Bins
<b>MSE</b>	Mean squared error
<b>Nadam</b>	Nesterov-accelerated adaptive moment estimation
<b>NKE</b>	Norrekær Enge I
<b>NN</b>	Neural network
<b>NSA</b>	Numerical sensitivity analysis
<b>PC</b>	Power curve
<b>PCA</b>	Performance Correlation Assessment
<b>PCV</b>	Power curve verification
<b>PCWG</b>	Power curve working group
<b>PDF</b>	Probability density function
<b>PReLU</b>	Parametric rectified linear unit
<b>PyC</b>	Pen y Cymoedd
<b>ReLU</b>	Rectified linear unit
<b>RMSE</b>	Root mean squared error
<b>SELU</b>	Scaled exponential linear unit
<b>TI</b>	Turbulence intensity
<b>TPPI</b>	Turbine power performance indicator
<b>TU Delft</b>	Delft University of Technology
<b>RBF</b>	Radial basis functions
<b>RH</b>	Relative humidity
<b>RMSprop</b>	Root Mean Square propagation

<b>SCADA</b>	Supervisory Control And Data Acquisition
<b>SGBRT</b>	Stochastic gradient boosted regression trees
<b>SGD</b>	Stochastic gradient descent
<b>SVM</b>	Support vector machine
<b>SVR</b>	Support vector regression
<b>UCL</b>	Upper control limit
<b>US</b>	Ultrasonic
<b>WICO</b>	WIND-consult GmbH
<b>WPC</b>	Warranted power curve

## List of Symbols

$\alpha$	Angle of attack [°]
$\alpha$	Distrubed sector angle [°]
$\alpha$	Wind shear component [-]
$\eta$	Learning rate
$\hat{y}_i$	Modeled output
$\lambda$	Regularization parameter
$\mu$	Mean
$\mu$	Momentum coefficient $\in [0, 1]$
$\rho$	Density [kg/m <sup>3</sup> ]
$\sigma$	Standard deviation
$\theta$	Blade pitch angle [°]
$\varphi$	Inflow angle [°]
$\varphi$	Wind veer [°/m]
$b$	Node bias
$c$	Blade element chord [m]
$C_0$	Unregularized cost function
$C_d$	2D profile drag coefficient [-]
$C_l$	2D profile lift coefficient [-]
$C_p$	Power coefficient [-]
$D$	Rotor diameter [m]
$dr$	Blade element length [m]
$E$	Expected value
$f(h)$	Activation function
$F_n$	Normal force [N]
$F_t$	Tangential force [N]
$h$	Sum of weighted inputs in a perceptron
$I$	Turbulence intensity [%]
$p$	Ambient pressure [hPa]
$p(u)$	Wind speed probability distribution
$R$	Rotor radius [m]
$Ri_b$	Bulk Richardson number [-]
$u$	Longitudinal wind speed component [m/s]
$v$	Lateral wind speed component [m/s]
$w$	Vertical wind speed component [m/s]
$w_i$	Connection weight
$y_i$	Measured output

---

# Preface

With the wind industry being a relative young industry there are still lots of approaches to pursue and processes to be optimized. One of which is combining machine learning with data from existing wind farms. The use of machine learning in this Master thesis was not included or mentioned in the initial scope of the project, this approach was my take on considering the options of approaching this project. This idea of machine learning was relative new to me, I only had a crash course in neural networks and evolutionary algorithms in my industrial internship at Siemens Flow Instruments. My supervisor Dietmar W. Weiss taught me a lot and introduced me to the interesting concept of machine learning. I greatly appreciate the time and patience sacrificed in order for me to learn about this topic, among others, and spark my interest.

I would also like to greatly thank Jan Coelingh, my supervisor at Nuon/Vattenfall, for -first of all- providing for the opportunity for me to do my MSc. thesis with Nuon/Vattenfall. But also thank you for guiding me through not only my thesis project, but also through a large company such as Vattenfall, introducing me to the different sides and helping me find my way in the Vattenfall documentation systems among others.

Furthermore I would like to thank Wim Bierbooms, my daily supervisor at the TU Delft who always had critical and supportive input, but also kept reading my updated report for our meetings, even though these were rather extensive in size.

A thanks to all my colleagues of Vattenfall, from the Wind Resource department, to the Data Team and R&D in Denmark. Thank you for having me as a part of the team, providing me for interesting, relevant input and supplying me with answers, insights and shared experiences.

I can look back at this period as an interesting period where I gained a lot of experience in the business side of wind energy and valuable insights in a large utility company, while gaining experience in machine learning and data analysis.

Finally, I would like to thank my friends and especially my parents for always being supportive, especially during more difficult times.

Delft, The Netherlands  
March 2, 2018

D.L. van der Arend



“The goal is to transform data into information, and information into insight.”

— Carly Fiori, President of Hewlett Packard 1999-2005



---

# Chapter 1

---

## Introduction

The wind energy business has experienced strong growth the last decade(s) and is expected to grow even more (Figure 1-1), due to the increased demand in renewable energy, the decrease of its cost and independence from fossil fuels. The latter is an ever increasing problem due to scarcity, (geo-)political instability, volatile markets and environmental problems. [28] However, while wind turbine size and installed capacity both keep sizing up and the wind industry keeps developing, wind as a resource still has to cope with a big intrinsic problem. Wind remains a stochastic energy source and is therefore difficult to predict, especially in the long term. But besides the behavior of the wind, there are other external factors affecting wind turbine performance, which can yield in deviations from performance predictions.



**Figure 1-1:** Global wind energy market forecast up to 2021 via Global Wind Energy Council [3]

It is this performance prediction which is a driving factor in the early phase of wind farm

development. The business case is build around some important aspects, among which the energy yield prediction over the wind farm lifetime is a leading number as it represents the large majority of revenue for the wind farm. This estimation is a driving force in placing bids in tenders for new wind farm locations; too high and the tender will not be won, while a too low estimation will result in lower revenue then expected, depreciating the investment.

Besides the long-term estimation of the wind turbine/farm performance, there is the point of product warranty. The turbine manufacturer delivers a product, the turbine, and promises a minimum performance. This warranted performance is verified for some time after installation and failure to meet the delivered Warranted power curve (WPC) has serious financial consequences. Careful verification of turbine performance is therefore an important process.

Despite the importance of the WPC in both these applications, it is still primarily based on wind speed with a single correction for Turbulence intensity (TI) and wind shear. This limits the flexibility of this WPC and in recent years this lack in flexibility has gained in interest. Some manufacturers currently provide two power curves, each for a certain range of atmospheric conditions. Furthermore the Power curve working group (PCWG) has been set up to research and evaluate new Power curve (PC) modeling techniques. [5] Research for new, alternative methods for modeling or refining the power curve are increased, not in the least because of the large growth of the industry.

## 1-1 Thesis motivation and research goals

While the warranted power curve method has its limitations in flexibility for external conditions, at the same time there is an ever increasing amount of data linking these external (atmospheric) conditions to turbine performance. It makes sense to use the latter to improve flexibility of the former and in the last decade multiple attempts have been done to improve this PC modeling through empirical data. In analyzing and application of large datasets, the machine learning paradigm is getting more interesting as these techniques have developed over the past decades and continue the rapid development. Therefore it should not come as a surprise machine learning techniques have been used in PC modeling with success. [1,2,20,29,30] While numerous examples are present how a turbine's power-velocity curve is modeled using machine learning, incorporation of multiple inputs and further application of such a model has barely been studied, despite the knowledge that multiple external inputs (like turbulence intensity or density) influence the turbine performance. [1,20,31–33]

Therefore the question arises, how can these databases of measured external (atmospheric) conditions, which can be linked to empirical turbine performance, be used to improve the insight these conditions have on turbine performance. The follow-up question is how this knowledge can then be used to improve current methods of long-term energy yield estimations, but also other processes like Power curve verification (PCV), where the warranted turbine performance is verified. For this the earlier mentioned machine learning field is used, as it is ideally suited for learning from provided data. This led to the main research question:

*How do external atmospheric influences individually impact wind turbine performance?*

## Research goals

While some influences are described in theory (sometimes extensively), others are not and these are generally not considered. Especially the latter parameters are of interest in this thesis. Furthermore, with this question (and answer) as backbone of this thesis, the follow-up question is how these found relations can be used to refine processes as the long-term energy yield estimation and power curve verification. This yields in three distinct research goals of this thesis, where the latter two build on the first:

- Improve insight in the trends or dependencies in influences of (site-specific atmospheric) parameters on wind turbine performance, via measured results and machine learning modeling.
- Study feasibility of a data-driven model, incorporating found dependencies, intended to refine long-term energy yield estimations for future wind farms.
- Study feasibility of a data-driven model, incorporating found dependencies, intended to refine power curve verification for individual wind farms and turbines.

## Approach

Since machine learning is getting increasingly advanced and is especially applicable to data analysis, it is a logical approach on achieving the thesis goals. The relevance will be further elaborated upon in the third chapter where also the underlying methodology or algorithm is discussed, but first a machine learning technique is chosen as the range of methods is extensive. This algorithm will then be adapted to the goals of this thesis and trained using datasets from several Vattenfall wind farms: Prinses Alexia, Pen y Cymoedd and Norrekær Enge I.

Using the adapted algorithm and the measured data from *Vattenfall* wind farms, the influence of selected external influences is singled out and dependencies modeled. These modeled dependencies can then be used to refine both the long-term energy yield estimation and the power curve verification on local atmospheric conditions. An important aspect in this is validation and verification of the model, since this model will not be a physics-based model (no set of physical equations). Current theory is not incorporated in an overarching model, since it is the impact outside this theory which is of interest. This brings along a risk of a lack of transparency of the model and results should therefore always be considered with care.

## 1-2 Thesis outline

The outline of this thesis report is as follows:

- Chapter 2 will first discuss the current method for turbine performance estimations. Here also the most important atmospheric influences on turbine performance, as found in literature, are summarized. Finally alternative machine learning methods for modeling the PC, which incorporate these influences, are introduced.
- Chapter 3 continues on a trade-off between discussed alternatives and continues on elaborating the working principles of the chosen machine learning method.
- Chapter 4 introduces the wind farms from which data was used for this thesis. The focus of this chapter is on wind farm Prinses Alexia.
- Chapter 5 shows the method of pre-processing the available data, filtering out noise and prepare the remaining data for application in the machine learning algorithm.
- Chapter 6 will then discuss model set up and the distinction into three different model applications is introduced.
- In Chapter 7 the validation and results of the three models and their applications are presented. Also limitations of these models are discussed.
- Finally the conclusions are presented in Chapter 8 and the thesis is closed of with recommendations for future work related to this thesis along with potential practical usefulness of the thesis models.

Further elaboration on the used wind farms, machine learning techniques and internal functions, sensitivity analyses, model optimizations and relations directly from measured data are presented in the appendices, at the end of this report.

# Wind turbine performance

While wind speed is the driving external parameter for turbine performance, other (external) atmospheric influences typically yield in some deviations from the warranted power curve. These deviations influences the accuracy of turbine performance predictions, which is elaborated upon in the first part of this chapter. For several of these parameters, some theoretical and/or empirical influences are very briefly described in the second part of this chapter. In the third part of this chapter alternative, machine learning based, methods for modeling the power curve are discussed. For more elaboration about these two topics, the reader is referred to the literature review prior to this thesis. [34]

## 2-1 Turbine performance prediction

Predicting the performance of a wind turbine combines several aspects from both the turbine and site conditions. Therefore it requires an accurate (long-term) environmental forecast, but also a power curve which represents the actual performance of the turbine. Relevant aspects are briefly discussed below.

### Estimation time-frame

There is a distinction in time-frame for turbine performance estimations. Short-term energy yield estimations are in terms of minutes up to days and primarily used for energy trading and grid management. Long-term predictions range over the lifetime of a wind farm and range from 20 to 25 years. These estimations are driving a business case and, in some cases, are used during operation to evaluate the farm or turbine performance. In this thesis only the long-term prediction is relevant, the short-term falls beyond the scope of this project.

### Current estimation method

The most common method for estimating the energy yields is described in the International Electrotechnical Commission (IEC) and is based on estimating the kinetic energy flux through

the rotor. [14] This kinetic energy flux  $P_{kin}$ , shown in Eq. (2-1), represents the effective wind power flux through the swept area of the turbine rotor. The rotor swept area  $A$  in  $m^2$  is integrated over, usually through discretization over the height of the rotor as wind speed and density change over altitude.  $\rho$  is the air density in  $kg/m^3$  and velocity  $u$  the perceived longitudinal wind speed in m/s averaged at the center of each discretized rotor area bin or, usually, at hub height. Only the longitudinal and lateral velocity components are taken into account.

$$P_{kin} = \int_A \frac{1}{2} \rho u^3 dA \quad (2-1)$$

The wind turbine only generates a certain factor of the total kinetic wind energy, indicated by the power coefficient  $C_p$  which is dependent on the tip speed ratio  $\lambda$ . Considering the Probability density function (PDF) of the wind speed at hub height  $p(u)$ , the rotor diameter as the area of a circle with radius  $R$  in meters and drive train efficiency  $\eta$ , the wind power equation can be reconstructed to Eq. (2-2). [35]

$$\bar{P}_w = \frac{1}{2} \rho \pi R^2 \eta \int_0^\infty C_p(\lambda) u^3 p(u) du \quad (2-2)$$

Wind speed distribution  $p(u)$  in Eq. (2-2) is typically a Rayleigh or Weibull distribution, where the latter is deemed slightly more more accurate. The Weibull distribution, shown in Eq. (2-3), assumes a shape parameter  $k$  and a scale parameter  $\lambda$  for the site. [36]

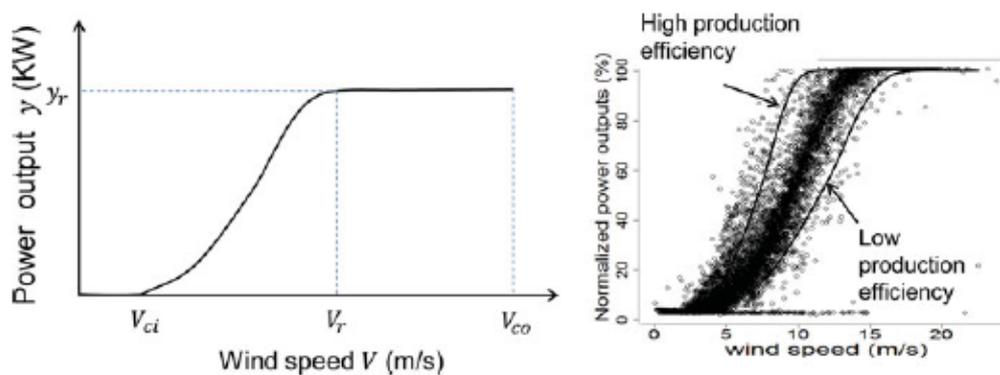
$$f(U) = \frac{k}{\lambda} \left( \frac{U}{\lambda} \right)^{k-1} e^{-\left( \frac{U}{\lambda} \right)^k} \quad (2-3)$$

While the wind speed and density are clearly present in the combination of Eq. (2-2) and Eq. (2-3), other (external) parameters such as Turbulence intensity (TI) are not taken into this equation, despite the effect of at least some -like TI and wind shear- has been shown in literature. [1, 4, 31, 32] The problem lies in the complexity of taking these into account. These parameters not only influence the turbine performance in a complex and typically non-linear way, they also influence each other yielding in a less predictable correlation. This makes modeling for all these parameters for a complex, highly non-linear model. Currently there is no such (commercial) model, capable of modeling for multiple atmospheric influences.

In determining the long-term energy yield prediction there are some other factors included. Additional losses can be attributed to wake effects, efficiencies, turbine and grid failure and turbine degradation. These are of a different kind of energy yield reduction and fall beyond the scope of this research.

## The warranted power curve

The performance of a wind turbine is quantified in a power curve, captured in the Warranted power curve (WPC) provided by the turbine manufacturer. This WPC indicates the performance for a range of wind velocities as *warranted* by the manufacturer. This is generally an averaged performance, as the real performance typically deviates from the WPC, as can be seen in Figure 2-1.



**Figure 2-1:** Wind turbine's power curve on the left, measured turbine performance on the right [4]

The WPC is constructed by the turbine manufacturer and can therefore be set somewhere in between conservative (warranting relatively lower performance) or very competing (relatively higher performance), which in turn influences the competitiveness of the wind turbine. This WPC is also provided with an uncertainty range (typically in the range of 3% - 6%), which indicates the uncertainty the turbine will not perform as warranted, due to various influences outside the power of the manufacturer. This WPC and uncertainty are driving performance indicators and key aspects in the choice for wind turbine in wind farms and are often negatively correlated. When a more conservative, or 'safe', version of the turbine's power curve is supplied by the manufacturer, the uncertainty is typically lower (as under performance is less likely to occur). But when the manufacturer decides to more or less *push* the warranted performance and provide for a better Power curve (PC), the uncertainty is typically higher. In a way the power curve does not show the actual performance of the wind turbine, but the performance as decided by the manufacturer.

There is a risk of making the WPC too competitive, however. Failing to meet the warranted performance, on average over a pre-determined timespan, within the set uncertainty has severe financial consequences. In contracts between turbine manufacturer and the buyer, clauses are described which define that the manufacturer has to compensate the farm operator (turbine buyer) if a turbine does not meet the warranted performance, within the uncertainty range. It yields additionally to reputation damage for the turbine manufacturer. So the PC determined and provided by the manufacturer always remains a trade-off between being competitive on the one side and risking under-performance on the other.

### Power curve working group (PCWG)

Nowadays a part of this uncertainty is captured by the introduction of an *inner* and *outer range* and typically for both ranges a different power curve is provided. The use of the inner and outer range has been proposed by the PCWG and by them proposed to the IEC as an improved method for determining the Annual energy production (AEP). This inner range is defined as a range of atmospheric conditions for which 100% AEP (relative to the power curve) is expected, while the outer range envelops the conditions excluded by the inner range. [37] This inner-outer range concept is visualized in Figure 2-2. The PCWG has a road-map set up for current and future proposals. [37]

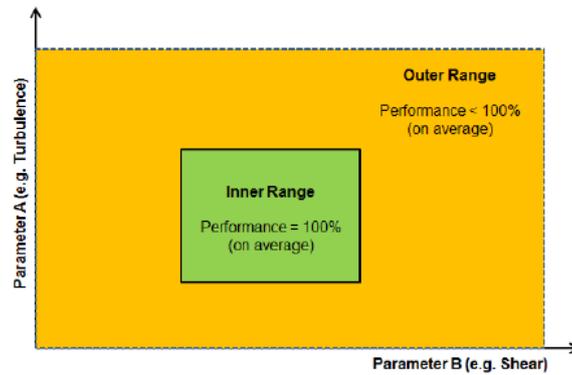


Figure 2-2: Schematic representation of the inner-outer range proposal of the PCWG [5]

## 2-2 Influences on wind turbine performance

From the power equation, Eq. (2-2), can be seen that besides the wind velocity, the density is the second major external influence on wind power and therefore turbine performance. Other parameters, however, are not directly visible in this equation while still (possibly) affecting performance directly or indirectly. Pelletier et al. describes the (indirect) influence on turbine performance and other external parameters, along with a visual aide as shown in Figure 2-3. In this section several (external) atmospheric variables and their influence on turbine performance are discussed.

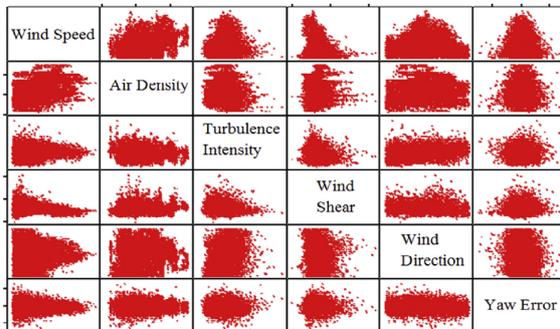


Figure 2-3: Example of measured influences on variables for 3-4 m/s bin. [1]

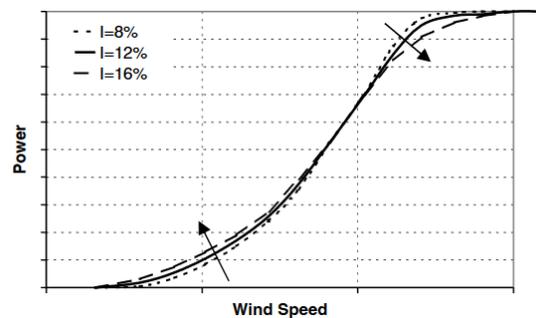


Figure 2-4: General effect of turbulence intensity on wind turbine performance. [6]

### Turbulence intensity

Several definitions for the Turbulence intensity (TI) exist in national and international standards, all resulting in more or less the same value of turbulence intensity (denoted by symbol  $I$ ). The definition used in this thesis uses the 10-minute averaged standard deviation ( $\sigma$ ) of the wind speed with the longitudinal mean wind speed  $\bar{u}$ , as shown in Eq. (2-4). [38]

$$I_u = \frac{\sigma_u}{\bar{u}} \quad (2-4)$$

The individual effect of turbulence intensity on wind turbine performance has been empirically studied. While no specific relation is defined, the effect are qualitatively defined. A general effect is shown in Figure 2-4. At lower wind speeds a higher TI has a positive effect on the performance. This is mainly caused by the steepness of the power curve at the start of the PC. Deviations which lead to an increase in wind speed have a larger positive effect on the performance than the negative effect of equal deviations resulting in lower wind speed. The reverse can be seen at the *knee* of the power curve; near rated power. Lower wind speeds due to turbulence have a stronger effect than higher wind speed, since the performance is flattened after reaching rated power. So higher winds speeds than average (due to high TI) barely lead to an increase in performance, while lower do reduce the performance noticeably.

## Wind shear

In assessing wind turbine performance the wind speed at hub height is typically taken. However, wind speed (generally) increases with altitude, which is modeled with a logarithmic or exponential (power) relation. For altitudes lower than 60m the logarithmic profile is assumed, while around 60m the relation changes to the power law as shown in Eq. (2-5). [35] In this thesis the power law is considered and quantified through the wind shear component  $\alpha$ .

$$u_i = u_{hub} \left( \frac{h_i}{h_{hub}} \right)^\alpha \quad (2-5)$$

With:

- $u_i$  - the longitudinal wind speed at altitude  $h_i$  in m/s
- $u_{hub}$  - the (longitudinal) hub height wind speed in m/s
- $h_{hub}$  - the hub height of the wind turbine in m
- $\alpha$  - the dimensionless wind shear component

In literature there is currently no consensus on the effect of wind shear on turbine performance. On the one hand, some researches conclude higher wind shear yields in higher performance due to higher wind speeds at higher altitudes. [39, 40] Other studies found no significant impact of wind shear on performance ([33]) or even yield in lower performance. [41, 42] The latter could be explained with the wind profile, where the gradient is much higher at lower altitudes than at higher altitudes (depending on the wind shear). Walter et al. found a parabolic impact of wind shear on performance deviations when studying both wind shear and wind veer, as shown in Figure 2-6, in the next section. [7]

Wind shear is highly related to atmospheric stability and therefore also TI. Multiple studies found the stable atmosphere tends to decrease performance with higher wind shear, while neutral and unstable atmospheres increased the efficiency for higher wind shear profiles. [43, 44] An explanation is proposed that the slope above the hub height is less steep in stable conditions, resulting in a more constant/linear wind velocity over the whole rotor. Wharton and Lundquist also looked into the effect of wind shear in combination with turbulence, via atmospheric stability. No significant shear effects were found in the upper half of the rotor,

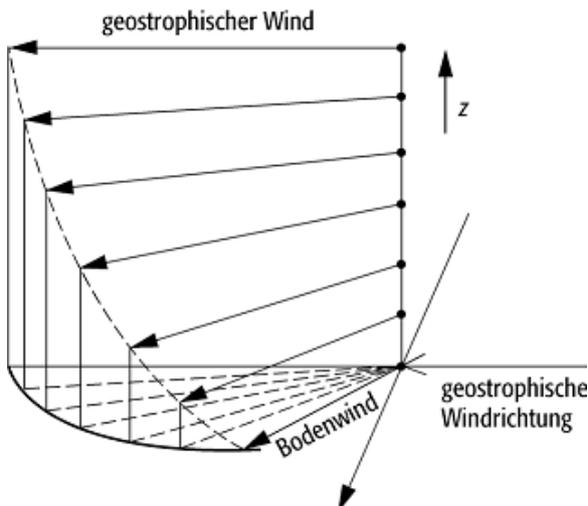
while the lower half of the rotor was leading. [33] This is in line with the explanation given by Dörenkämper et al. indicating the steeper the slope, the lower the influence of the wind shear (as wind speed increases less per unit length). [44]

## Wind veer

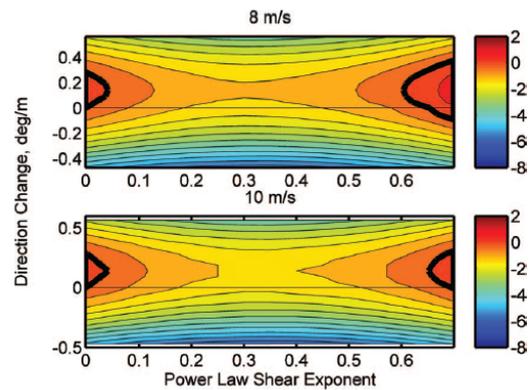
Wind veer, the change of wind direction for altitude, is becoming increasingly relevant as turbine rotors increase in size. Wind direction tends to change with altitude since pressure fields are not vertical columns in the atmosphere. Irregular shapes naturally yield in non-constant wind direction over altitude. The Ekman spiral is the most known structure occurring in the lower atmosphere (Figure 2-5). A variation in wind direction over altitude has direct influence on the inflow angle and therefore differences in lift and drag forces which reduce efficiency. Besides a negative effect on the performance of wind turbines, it also yields in higher load variations and therefore increased fatigue damages. The wind veer is assumed to be linear with altitude, therefore having a relation as shown in Eq. (2-6).

$$\text{Wind veer} = \frac{\Delta(\text{Wind direction})}{\Delta(\text{Altitude})} \quad (2-6)$$

While the research on this topic is very limited, Walter et al. looked into the combined effect of wind shear and wind veer on the performance of the wind turbine. [7] They found a negative influence in more or less all conditions, except for very low or very high wind shear components (Figure 2-6). Higher (absolute) wind veer tends to decrease performance, which makes sense due to the sub-optimal inflow angle for the full rotor. Effects of the inflow angle are discussed next.



**Figure 2-5:** Ekman spiral visualized (via techniklexikon.net)



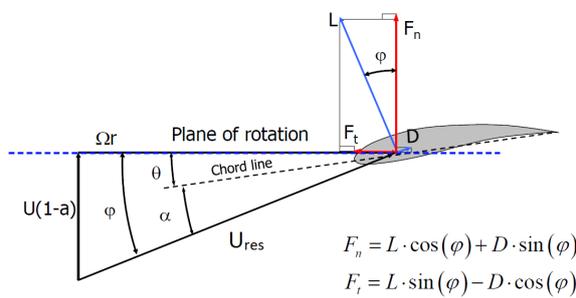
**Figure 2-6:** Percentage of change in power production (scale on the right) for a range of wind speed and direction shear, using hub height wind speed of 8 m/s (top) and 10 m/s (bottom). Power change interval is 0.5%, the thick black line represents a zero change contour. [7]

## Inflow angle

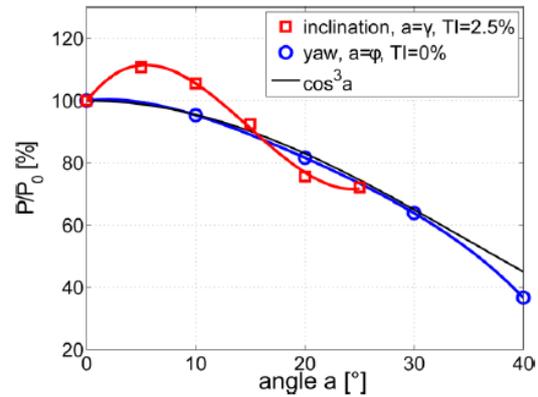
The inflow angle,  $\phi$ , affects turbine performance through 2 scopes. On a larger, 3D scale the inflow angle affects the longitudinal wind speed  $u$ , experienced by the rotor. Would all three wind speed components be measurable (with an Ultrasonic (US) anemometer for example), the inflow angle  $\phi$ , relative to the axis of rotation, is calculated with Eq. (2-7), using wind velocity magnitudes  $u$ ,  $v$  and  $w$  in longitudinal, lateral and vertical directions. A larger inflow angle means the longitudinal wind speed magnitude is smaller relative to the absolute wind speed magnitude.

$$\tan(\phi) = \frac{w}{\sqrt{(u^2 + v^2)}} \quad (2-7)$$

On a smaller scale Blade element momentum (BEM) is used, which further explains the impact on turbine performance. Here a typical cross-section of a wind turbine blade is used, as shown in Figure 2-7, where incoming wind speed  $U$  represents the longitudinal wind speed  $u$ . The blade is pitched at a certain angle  $\theta$  and the ratio between the incoming longitudinal wind speed and rotational speed (of the element) determines the angle of inflow  $\varphi$  (different from  $\phi$ ). From this the Angle of attack (AoA),  $\alpha$ , is deduced. This AoA translates to a certain lift coefficient  $C_l$  and drag coefficient  $C_d$ .



**Figure 2-7:** Blade element and its perceived wind speed with resulting forces (via lectures Michel Zaaijer at TU Delft).



**Figure 2-8:** Comparison of power production for a yawed and tilted. Note that yaw angle  $\varphi$  is different from blade element inflow angle  $\phi$ . [8]

The lift and drag forces of a blade element (respectively equations (2-8) and (2-9)) are a function of the normal force  $F_n$  and tangential force  $F_t$ , Eq. (2-10), where  $F_t$  is the driving force for mechanical energy in the turbine. Relating this to the lift and drag forces yields in Eq. (2-11).

$$dL = \frac{1}{2} \rho c_l U_{res}^2 c \cdot dr \quad (2-8)$$

$$dD = \frac{1}{2} \rho c_d U_{res}^2 c \cdot dr \quad (2-9)$$

$$dF_t = dL \cdot \sin(\varphi) - dD \cdot \cos(\varphi) \quad (2-10)$$

$$dF_t = \frac{1}{2} \rho U_{res}^2 c \cdot C_l \left( \sin(\varphi) - \frac{\cos(\varphi)}{\frac{C_l}{C_d}} \right) dr \quad (2-11)$$

Where:

- $dF_t$  - Element tangential force in N
- $U_{res}$  - Resultant velocity in m/s
- $c$  - Blade element chord in m.
- $dr$  - Length of blade element in m
- $C_l$  - Profile lift coefficient [-]
- $C_d$  - Profile drag coefficient [-]
- $\varphi$  - 2D blade element angle of inflow in degrees/radians

As can be seen in Eq. (2-11), the factor  $\frac{C_l}{C_d}$  has to be maximized to optimize the tangential force for certain wind speed and blade element inflow angle  $\varphi$ . Since both lift and drag coefficients are direct functions of the AoA, which is affected by the blade element inflow angle, a sub-optimal blade element inflow angle  $\varphi$  results in a decrease in tangential force and therefore mechanical power.  $\varphi$  was a direct result from incoming wind speed  $U$ , which is longitudinal wind speed  $u$  on a larger scale. And with  $u$  decreasing with higher inflow angle  $\phi$  (for constant wind vector), higher inflow angles reduce are expected to reduce the turbine performance.

Tsalicoglou et al. simulated the effect of the inflow angle  $\phi$  on the turbine performance [8]. By yawing and tilting the rotor, certain inflow angles are induced and the performance is simulated. Their simulations show an increase in performance for small inclination angles up to about  $12^\circ$  as can be seen in Figure 2-8. In practice the rotor is sometimes already tilted slightly to increase the blade-tower clearance. The graph also shows the yaw angle reduces performance with the cubed cosine of the angle, which is a direct result of the loss in longitudinal velocity, in the power equation (Eq. (2-2)).

## Relative humidity

Relative humidity (RH) is the percentage of water vapor in the air. Since the density of water vapor is different than dry air, RH directly influences the humid air density  $\rho_h$  through Eq. (2-12) (via [45]). The effects of density is discussed in the next section.

$$\rho_h = \frac{1}{T} \left( \frac{p}{R_{dry}} - RH \cdot p_{vapor} \left( \frac{1}{R_{dry}} - \frac{1}{R_{vapor}} \right) \right) \quad (2-12)$$

in which:

- $T$  - Temperature in Kelvin.
- $p$  - Ambient pressure in hPa.

- $R_{dry}$  - Specific gas constant for dry air ( $\approx 287.05$  J/kg/K).
- $p_{vapor}$  - Pressure of water vapor, calculated with Eq. (2-13) or Eq. (2-14).
- $R_{vapor}$  - Specific gas constant for water vapor ( $\approx 461.495$  J/kg/K).

$$p_v = RH \cdot 6.1078 \cdot 10^{\frac{7.5T}{T+273.3}} \quad (2-13)$$

$$p_v = 0.0000205e^{0.0631846 \cdot (T+273.3)} \quad (2-14)$$

Where temperature  $T$  is in °C for Eq. (2-13) and Eq. (2-14).

The effect of humidity on the performance has barely been studied, except through the influence of density. It has, however, recently been simulated by Yue et al. by means of Computational fluid dynamics (CFD) [46]. By simulating a wind turbine in *Fluent* in relative dry and humid conditions, the power deviations were obtained. They found especially the combination of high humidity and high temperature have a noticeable effect on the AEP. A difference of up to 1.92% in wind power could be monitored between dry air and 90% humidity at 35°C.

Humid air has another potential problem for wind turbines. Due to higher humidity a water film can form on the blades, which alone is a contamination of the airfoil. But additionally this film of water can lead to dust and insects sticking to the blade, but also to icing on the blade. While these effects can barely be quantified and are a separate study on its own, performance blade degradation due to contamination could be noticed in measured performance.

## Density

The density of air,  $\rho$ , is the mass per unit volume in kg/m<sup>3</sup>. While the ISO air density is set at 1.225 kg/m<sup>3</sup> at 1 bar and 15°C, the density is also directly calculated from the temperature, pressure and humidity of the air. To calculate the density from these factors Eq. (2-12), described above, is used [45]. The density has a direct linear influence on the turbine performance through the power equation, Eq. (2-2). Furthermore the IEC specifies the wind speed  $U$  has to be corrected for the density, using Eq. (2-15). [14, 38] In practice this means the power curve is corrected for the average measured density on site.

$$U = U_0 \left( \frac{\rho}{\rho_0} \right)^{\frac{1}{3}} \quad (2-15)$$

with:

- $U_0$  - Bin wind speed in m/s
- $\rho_0$  - The reference density for which  $U_0$  is valid (ISO = 1.225 kg/m<sup>3</sup>)

## 2-3 Power curve modeling using machine learning

Besides the more 'traditional' method of modeling a power curve, the use of machine learning has been studied several times in the past. Instead of using the power curve and the Method of Bins (MoB) from the IEC, data obtained from operating wind turbines or wind turbine simulations were used in a machine learning algorithm to model a generalized power curve. This is obtained from using existing data and training the model/algorithm out of this data. It has been applied successfully in previous researches usually for one parameter (velocity), although in increasing studies with multiple parameters. In this section several methods are compared and briefly discussed, especially from comparing studies. This yields in a short overview of several interesting methods. A comparison and choice for method is performed in Chapter 3, where this method is also discussed in more detail. For a more extensive description of these alternative methods of PC modeling, the reader is referred to the literature review performed prior to this thesis. [34]

### Parametric modeling

In machine learning there are initially two distinct methods: parametric and non-parametric modeling. The former, parametric, means *a priori* knowledge is required and a set of equations has to be set up. These equations contain some unknowns which then have to be solved so that the complete model fits the training data best (through optimization of a loss function). This results in a white box model, where one knows what is going on inside the model and can extract information out of it. Typical methods in parametric modeling are:

- Linearized segmented modeling
- 4 Parameter logistic function
- 5 Parameter logistic function
- Polynomial function

Linearized segmented modeling means discretization the expected model in segments, consisting of linear functions. 4 and 5 parameter logistic function means setting up the a power function dependent on 4 or 5 parameters, which represents the problem (PC in this case). The unknowns then have to be solved to the training data. Eq. (2-16) is a typical example of a 5 parameter logistic function. [47, 48]. Finally a polynomial function described the problem as a polynomial, for which the coefficients have to be solved (Eq. (2-17) [25, 47])

$$P = f(U, \theta) = d + \frac{a - d}{\left(a + \left(\frac{U}{c}\right)^b\right)^g} \quad (2-16)$$

$$P_i = \beta_0 + \beta_1 b U_i + \beta_2 U_i^2 + \dots + \beta_k U_i^l + \varepsilon_i \quad (2-17)$$

These methods all require functions to be defined, which describe the model. From this internal function parameters have to be optimized to fit the model to the training data.

Most of these methods still have just one physical parameter ( $U$ ), but at least two internal parameters (in parametric vector  $\sigma$ ) for each function. As adding more physical parameters drastically increases model complexity, one can imagine how a model would look like with six physical parameters and a proportional amount of internal parameters. Janssens et al. state that modeling with multiple input variables would be nearly impossible. [2]

Even though parametric models incorporate the theory in a model, they have some clear disadvantages. They require sufficiently quantified a priori knowledge which has to be translated into a model with a lot of parameters, both physical and internal. This a priori knowledge also limits the model itself, as it usually results in constraints and predetermined behavior of the model. Not in the least, the influence of several parameters on each other and the turbine performance is not necessarily 100% known or there might not be consensus (for example the influence of wind shear [33]). These reasons, together with the thesis goal of incorporating multiple physical variables in a model, indicate that parametric modeling will not be the most suitable method for this thesis. It would seem a model less dependent on a priori knowledge would be better.

## Non-parametric modeling

The opposite method of parametric modeling is non-parametric modeling which, as the name suggests, does not involve finding internal parameters in a set of equations. In fact, typically no a priori knowledge about the underlying physics is required at all, as the idea is to find (statistical) dependencies between input and output without any constraints from previous theory or functions. Of course validation and analysis of the results does require knowledge about the underlying physics. One can describe this as having potentially unlimited parameters, as the machine (the algorithm) tries to find a solution in the infinite parameter space in which all provided data points exist. This does require a large amount of data points from which the algorithms optimizes the solution. Non-parametric models are typically black-box models and it is therefore unknown how the internal model works (except the internal algorithms mechanics). Therefore no intrinsically useful information about the model can be extracted from it. While an enormous amount of non-parametric models exist, the ones evaluated here are limited to the Artificial neural network (ANN), decision tree, k-Nearest neighbor (k-NN), Support vector regression (SVR) and clustering fuzzy logic, which are methods used in comparing studies for PC modeling.

In PC modeling there have been several successful applications using non-parametric methods. Most of these have only used limited input parameters, a large majority only one (velocity). As early as 2001 a regression model was compared to a neural network in univariate PC modeling (single input, single output) and the neural network outperformed the 2<sup>nd</sup>, 3<sup>rd</sup> and 4<sup>th</sup> order regression models. [49] In later years there have been several studies comparing both some parametric as well as non-parametric models with each other in modeling the power curve, usually univariate modeling (single input) for wind turbine monitoring, performance estimation and power curve verification purposes.

In the following years various methods were tested for modeling the PC, typically for only a single parameter. Comparing studies with different parametric and non-parametric models showed the 5-parameter logistic function to perform best, while neural networks performing best of the non-parametric models. [47, 48, 50] However, this conclusion was not

always obtained and k-NN was sometimes found to perform best, while the Neural network (NN) performed worst. [51] But as the amount of input variables increased the k-NN performance degraded rapidly, while others (Cluster center fuzzy logic (CCFL), Multi-layer perceptron (MLP) and Adaptive neuro-fuzzy interference system (ANFIS)) improved in accuracy shown in Table 2-1. [20]

**Table 2-1:** Model results for univariate and multivariate PC modeling [20]

	Single input parameters		Three input parameters	
	MAE	RMSE	MAE	RMSE
CCFL	1.94	2.78	1.68	2.42
MLP (NN)	1.94	2.78	1.62	2.34
k-NN	1.95	2.80	6.02	5.96
ANFIS	1.94	2.78	1.60	2.30

The past few years three studies in multivariate PC modeling found good results for some non-parametric models, also compared to parametric models. [1, 2, 20] In these studies the MLP, a type of NN, and ANFIS were methods for which modeling results were most accurate, while regression trees proved to be very suitable on determining variable importance in PC modeling. [2]

## 2-4 Discussion on power curve modeling

This Chapter started with some elaboration of various external, atmospheric parameters which (might) influence the power production of a wind turbine. The most important parameters, which are taken into account in current models/methods, are the TI, density and wind shear. The relations found in literature will be important to verify modeled results.

Two alternative ways of modeling the power curve have been elaborated upon: parametric and non-parametric. The parametric method requires a priori knowledge, (likely) resulting in a white box method, but is also constrained by the functions defined in the model. Furthermore, it becomes increasingly complicated to model wind turbine performance for multiple inputs, considering input correlations (which have not necessarily been defined in theory). On the other hand there is non-parametric modeling where an advanced algorithm is used to find dependencies between provided training data. No theoretical constraints are attached to this method, where accuracy is typically limited to model layout and settings.

The goals of this thesis are to gain more insight in the influence of multiple parameters on turbine performance and study feasibility of using the chosen method in multivariate power curve modeling. While a parametric model can possibly handle a few inputs, the amount of intended inputs (being 7) required the model to be highly complex and non-linear. Furthermore, it would be too constrained by theoretical functions, used in the base of the model. Non-parametric modeling requires a lot of data and tries to find dependencies between input parameters and output (normalized turbine performance, or performance change). Considering these goals and results from previous research the non-parametric method is chosen to continue with. But as there is a vast amount of different methods within non-parametric modeling, the most suited algorithms has to be chosen. This is discussed in the next chapter.

---

## Chapter 3

---

# Machine learning

The base method for this thesis is non-parametric regressive machine learning, which in essence means providing sufficient data to a designed algorithm -the machine- to let the algorithm(s) deduct or optimize a model. Internal parameters are optimized to fit the data as best as possible. Basically the algorithm learns from examples, just as a human being learns by doing or seeing, hence the name machine learning. This technique is not new, but has seen rapid development in the last decade(s) especially in terms of new or improved algorithms fitted for the rapid increase in computational speed. Currently there is a vast amount of different algorithms, each with its own method and application. A suitable algorithm was chosen in the literature review which was performed prior to the start of this thesis. The trade-off will be briefly discussed first in this chapter. An important practical note is that the author of this report had some, but limited, knowledge regarding machine learning prior to the start of this thesis. The decision was therefore made to use existing software with these algorithms, excluding highly complex methods.

After the brief trade-off discussion in Section 3-1 on the method to continue with, the algorithm is elaborated upon in more detail in Section 3-2. Other methods are briefly described in Appendix D if the reader is interested in this topic. Application and modeling with the chosen method is discussed in Chapter 6.

### 3-1 Method trade-off

A variety of machine learning methods were compared in power curve modeling. A small selection from the mentioned literature review is compared here, consisting of: Neural network (NN), Support vector regression (SVR), Fuzzy logic and regression trees. For this a multiple criteria analysis was done to quantify a range of requirements, both practical as well as performance of the method. The analysis is shown in Table 3-1, together with the weight of each criteria. Requirements are:

- Availability - Is this method (readily) available as software package for (preferably) Python? In other words, how much of the machine learning algorithm still has to be written?
- Ease of use - How difficult and time consuming is making a model out of the given method. Both in user-friendliness as well as construction of a model out of it. Does it require a lot of additional input, or is it quite readily available for training a model?
- White-box approach - How well can the model internals be adjusted and are the model internals useful? Lower ranking means more of a black-box approach.
- A priori flexibility - How much of the model has to be defined from existing knowledge and is therefore also limited to a priori knowledge? Higher marks mean less required a priori knowledge and therefore potential new relations can be identified.
- Small dataset - How well does the method work with a small dataset? Should the dataset be small, how well does the method perform?
- Large dataset - How well does the method work with a large, possibly ever expanding dataset? Should the database be extensive and possibly be extended each month (for example), how well can the model cope with this?
- Human interference - How much human interference is required for the model to work and keep running, excluding starting the algorithm or model?

**Table 3-1:** Multiple criteria analysis trade-off between four selected methods for machine learning regression

Requirement	Weight	NN	SVR	Fuzzy logic	Regression trees
Availability	5	5	5	2	3
'Ease' of use	3	4	4	2	2
White-box approach	2	1	2	3	4
A-priori flexibility	3	5	5	2	1
Small dataset	3	2	4	5	5
Large dataset	4	5	1	5	5
Human interference	3	5	5	2	2
Sum		<b>95</b>	90	69	73

The neural network and support vector machine come out best in this analysis, from which the neural network seems to be the best choice. Both methods are black-boxes, widely available as software packages. The main difference between these two is the 'ideal' dataset size range; Support vector machine (SVM) can handle small datasets very well, contrary to NN. Due to the large amount of internal parameters of a NN, small datasets require smaller, less complex networks. On the other hand, a NN performs increasingly better for (increasingly) large datasets, while SVM generally starts having trouble for datasets over a couple of thousand samples. Therefore the neural network will be the base of the model. How a neural network works will be briefly explained next.

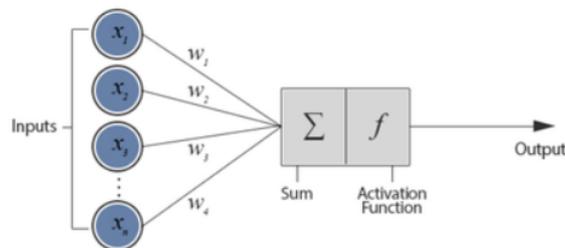
## 3-2 Artificial neural network

A neural network, or Artificial neural network (ANN), is a mathematical model which finds its base in the workings of the (human) brain. The brain consists of a large set of neurons/nodes, which are connected to and interacting with each other; the nervous system. In a NN, nodes and connections also form the basis and enable the network to model complex relations. Such a network is a full *black box* approach as there is barely any control of the model except through the (external) hyper-parameters. A trained network also provides barely any useful intrinsic information, from the workings of the model itself no useful information can be obtained as the internal parameters have little to no meaning outside of the model. In this section the basics and relevant working of an ANN are discussed.

### Single perceptron

The most simplest form of NN is called a *perceptron*, which is a single node or neuron. Figure 3-1 shows an overview of what this perceptron does. This perceptron takes in a certain amount of inputs ( $x_i$ ) from  $N$  input nodes which are connected to the perceptron, where each connection has a weight  $w_i$ . The input parameters and weights have numerical values in this application of regression. At the node (gray boxes in Figure 3-1) a sum  $h$  of all input nodes and their connection weights is performed, using Eq. (3-1). [9]

$$h = \sum_{i=1}^N w_i x_i \quad (3-1)$$



**Figure 3-1:** Inside a neural network node (via [www.datacamp.com](http://www.datacamp.com))

The other half of the node exists of an activation or transfer function  $f(h)$ , which has to be defined by the user. This function processes the combined inputs and provides for an output. An example of a sigmoid logistic function is given in Eq. (3-2). Activation functions introduce non-linearity in the network if sufficient layers are used, this is discussed later in this Chapter.

$$f(h) = \begin{cases} 0, & \text{if } h < 0 \\ \frac{1}{1+e^{-h}}, & \text{if } h \geq 0 \end{cases} \quad (3-2)$$

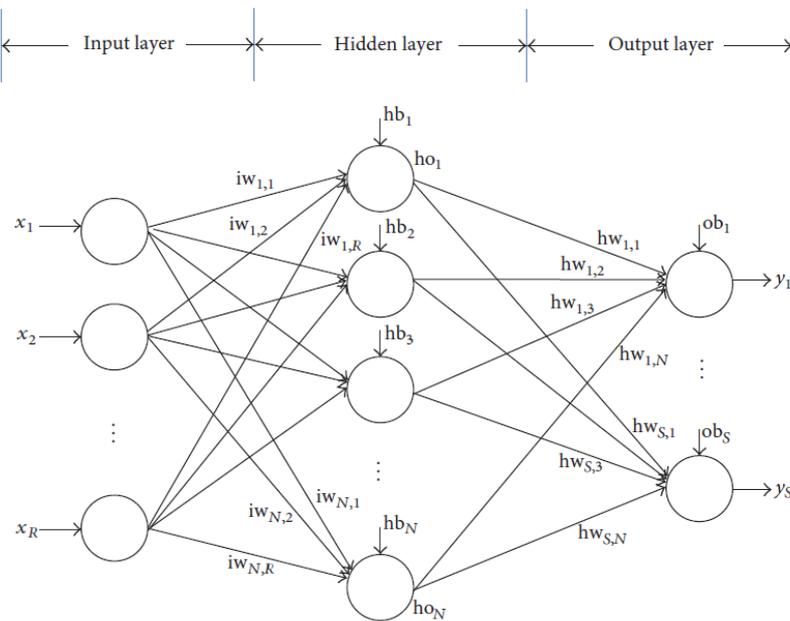
Additionally, the node itself has a bias ( $b$ ), which gives additional weight to the output of the node. This bias allows for shifting the activation function and directly influences  $f(h)$ ,

for example through the steepness of the activation function. Simply stated, in the linear function  $y = ax + b$ ,  $b$  is the bias. Finally, the output of the activation function is passed as output  $y$  of the node itself; Eq. (3-3).

$$y = f\left(\sum_{i=1}^N w_i x_i + b\right) \quad (3-3)$$

## Network architecture

The above description is for a single perceptron, but one perceptron yields in a poor model. To increase the flexibility and complexity of the neural network, several perceptrons are put into a *layer* and several layers are placed behind each other. This yields in a Multi-layer perceptron (MLP), the most widely used network architecture. In the MLP each node in a layer is connected to every node in the previous and next layer, if present. As described with the perceptron, each of these connections has a weight  $w_i$ . A simple overview of this architecture is shown in Figure 3-2. These networks typically consist out of three different kinds of layers: the input layer, hidden layers and the output layer.



**Figure 3-2:** Simple feed-forward (artificial) neural network architecture (network topology) with a single hidden layer [9]

Input is provided at the input layer and propagates through the hidden layers to the output layer, where the output  $y_i$  is read; this architecture is called a *feed-forward* neural network. Information is only fed forward in the network. More complex architectures may include connections skipping layers, cycles within the network or input entering in a later layer. These are, however, more difficult to design and are not considered for this thesis; only feed-forward networks are considered.

### Input layer

The input layer is the first layer of a network and gets 'raw' input parameters ( $x_i$ ) as input. Each node receives an input and is (typically) connected to each node in the next layer with weight  $w_i$ . Input nodes do not have their own activation function or bias.

### Output layer

The output layer is another base layer and is located in the end of the network. Instead of passing the output of this layer as input  $x_i$  to a consecutive layer, the output is directly read as model output  $y_i$ . This output is directly read out of the model and used to evaluate the fitness of the model (how well it performs to the error or loss function).

### Hidden layers

Between the first and last layers there can be an almost unlimited number of hidden layers, which consist out of a (pre-)determined number of perceptrons. These layers are called hidden layers since they can not be interacted by directly by the user, except through setting the hyper-parameters, and therefore are hidden from the user. Subsequent layers do not necessarily need to have the same amount of nodes. It is not always better to have a high number of nodes and layers, as this increases required computational power and does not necessarily result in a more accurate model as will become clear. More layers and nodes also increases the difficulty for training as there are more unknowns, which poses a problem especially for problems without a large dataset.

## **Activation function**

The activation function in a node takes in the combined weighted input from all previous connections, applies the function to this value and transfers the output as output of the node, which is then connected to each node in the next layer. While no rules or absolute truths exist for choosing the activation function, there are certain disadvantages on some activation functions, the most important ones are:

- *Saturation and gradient killing* - Some activation functions (like sigmoid or TanH) saturate at their 'ends' to -1, 0 or 1 (in other words; the gradient nears 0 for high function input). In gradient based learning (e.g. back-propagation), this gradient is multiplied with the node output gradient and with very low gradients at both ends the output is near 0, 'killing' the node. Therefore sigmoid or TanH functions are rarely used anymore in perceptron layers.
- *Killed threshold function* - Another risk is when using threshold function (only activates if  $h > 0$ ). During training of the networks some connection weights could develop such that the activation function is never activated and also kills the node.

Activation functions are defined for all nodes within a layer, excluding the input layer. The output layer activation function depends on the type of expected output as the output should fit well in the range of the activation function. As for the hidden layers, there is again no rule which sets the optimal function. However, one has to keep in mind as stated earlier. In recent years the Rectified linear unit (ReLU) had been popular, but as it sometimes 'kills' nodes, other functions based on ReLU were developed. Exponential linear unit (ELU) and Parametric rectified linear unit (PReLU) are attempts to fix this problem with ReLU. For this reason they have grown rapidly in popularity. [52] In the constructed network the ELU function is chosen for the hidden layers as the problem of node killing is barely present, while a certain degree of non-linearity is maintained.

### Loss/error function

The loss function, also known as cost or error function, indicates a statistical error between training and modeled data and is the function to be minimized as is standard in optimization problems. For the neural network this means optimizing the internal and hyper-parameters such that the loss function is minimal. Typical error functions are Mean squared error (MSE), Mean absolute error (MAE) and Mean absolute percentage error (MAPE) among others, shown respectively in Equations (3-4) to (3-6). The used loss function will be MAE.

The reason for using MAE is that for MAE value of each error contributes equally, no matter the magnitude of the error itself or relative to the measured output. For MSE the error becomes less important for lower values, due to the quadratic function. And with MAPE the error is weighed relative to the measured output, so equally large errors for low measured power are more important than the same error for higher measured output. In this analysis the magnitude of the error is less important (so MSE is less attractive) as is the relative error (so MAPE is less attractive), which results in MAE as error function.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (3-4)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (3-5)$$

$$MAPE = \frac{1}{N} \sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{y_i} \cdot 100\% \quad (3-6)$$

Where:

- $y_i$  - Training (measured) output for data point  $i$
- $\hat{y}_i$  - Modeled output for data point  $i$
- $N$  - Amount of data points

## Training / optimization

One of the most important, and the most time consuming, aspects of a neural network is the training process. Training means optimizing the internal weights  $w_i$  and biases  $b_i$  to fit the data to a generalized model. This is done by feeding the training data through the network and updating the internal parameters ( $w_i$  and  $b_i$ ) to fit input to output by means of the loss or error function. This is typically done in batches, with a batch being a subset of the training dataset. A single iteration is defined as one batch propagating through the network, while one epoch is defined as the full dataset propagating through the network. The training process goes on until a number of pre-defined epochs have been reached or the defined error function is sufficiently converged. Training of a NN requires a large database, the minimum size being very dependent on the amount of internal parameters (or vice versa). One rule of thumb is to have at least 30 times more data samples than unknowns in the network to consistently find the global minimum. Having too few samples is likely to result in oscillation in optimizing and over-fitting.

There are a multitude of optimization functions, which can be divided in three variations: Batch gradient descent (BGD), Stochastic gradient descent (SGD) and Multi-batch gradient descent (MBGD), where MBGD is a combination of the first two.

### Batch gradient descent

BGD is the 'standard' gradient descent function, updating the internal parameters  $x_i$  by calculating the gradient of the loss function to each internal parameter ( $\frac{\partial f(x_i)}{\partial x_i}$ ) for the training dataset. This is done using Eq. (3-7), where  $\eta$  represents the learning rate. Since the parameter gradients are to be computed for the whole dataset for a single update, this can become quite slow and prone to memory problems. The learning rate determines the size of the iteration or update and therefore partly the speed and accuracy of the optimization step. BGD typically converges to a local minimum which is not necessarily the global minimum. BGD is typically not a good choice for machine learning and/or large dataset sizes.

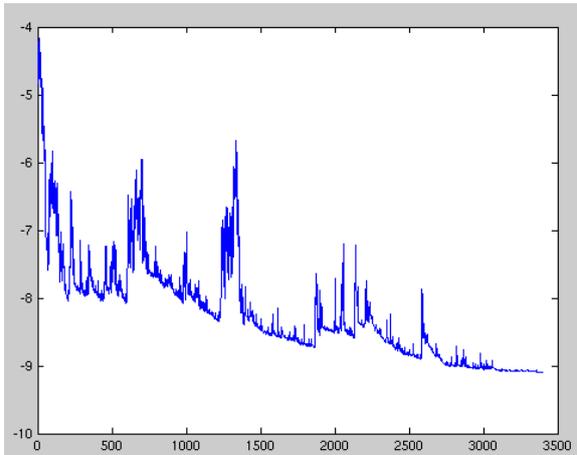
$$x_i = x_i - \eta \frac{\partial f(x_i)}{\partial x_i} \quad (3-7)$$

### Stochastic gradient descent

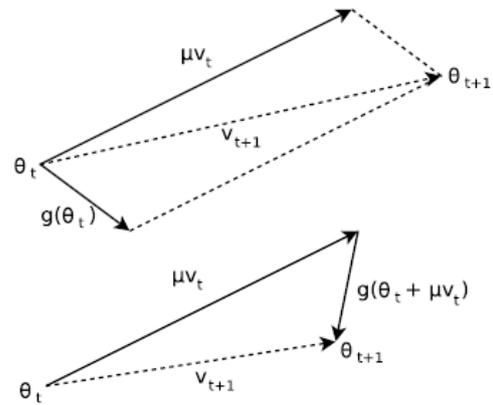
SGD functions do not update after the complete dataset, but after each training example so updating the internal parameters occurs much more frequent. This also yields in large fluctuations in the loss function evaluation as the full dataset is propagated through, for example as shown in Figure 3-3. This fluctuation, however, causes a certain stochastic behavior of the optimizer and enables it to jump out of local minima, to better locations. The downside of this behavior is that converging to the global minimum might be difficult and take long due to this overshooting of the minimum. This can be solved by lowering the learning rate, so called *decay*. This makes it very suitable for both convex and non-convex problems.

### Multi-batch gradient descent

MBGD combines both methods by performing an update for every mini-batch (instead of every every full batch or sample). This yields in a more stable convergence as the parameter updates are closer to each other, resulting in less 'spiky' behavior, compared to SGD. But it also uses optimization matrices which makes gradient computation for mini-batches very efficient. MBGD is typically the go-to for neural networks. However, this method does come with some challenges in choosing a proper learning rate and learning rate decay. A learning rate which is too high will not converge and if the decay is too small the decay has barely any influence for example. [10]



**Figure 3-3:** Fluctuations in training using SGD (via [10])



**Figure 3-4:** Top: classical momentum method. Bottom: Nesterov accelerated momentum [11]

On top of the 'standard' mini-batch gradient descent momentum can be added to accelerate the optimization process. The principle was introduced by Polyak and is a technique which adds momentum (velocity) to the gradient descent and therefore accelerates the training or optimization of an objective function. [53] In a NN this translates into training of the network. Classical momentum determines the gradient for the next iteration using Equations (3-8) and (3-9). [11]

$$v_{t+1} = \mu v_t - \eta \nabla f(\theta_t) \quad (3-8)$$

$$\theta_{t+1} = \theta_t + v_{t+1} \quad (3-9)$$

where:

- $v_t$  - Optimization velocity vector for iteration  $t$
- $\mu$  - Momentum coefficient  $\in [0, 1]$
- $\eta$  - Learning rate ( $> 0$ )
- $f(\theta)$  - Objective/loss function, to be minimized

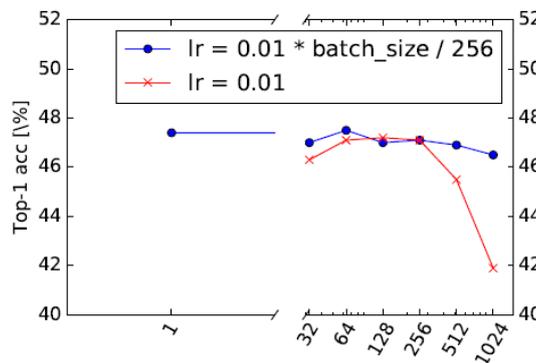
- $\nabla f(\theta_t)$  - Gradient at  $\theta_t$

Nesterov's accelerated gradient adds to this set of equations a slight change in the gradient of the objective function. Instead of just adding momentum to the (negative) gradient and learning rate ( $-\eta\nabla f(\theta_t)$ ), the momentum is also put within the gradient. This results in a new velocity vector (Eq. (3-8)), as shown in Eq. (3-10). How this method differs from the classical method is visualized in Figure 3-4. This method has shown to be faster in convergence rate with respect to the classical momentum method.  $\nabla f$  from Equations (3-10) and (3-8) is represented as  $g$  in Figure 3-4.

$$v_{t+1} = \mu v_t - \eta \nabla f(\theta_t + \mu v_t) \quad (3-10)$$

## Batch size

The MBGD optimizer uses a pre-defined batch size of samples for which to update the internal parameters in one iteration. Small batch sizes tend to take a lot of time (as a lot of iterations are required), but also zig-zag along the solution space as the gradient for small batches is likely to be very different from batch to batch. Very large batches, on the other hand are fast but can lead to under-fitting as the steps in the solution space are typically large and the optimizer can easily overshoot local minima, especially the global optimum, if the learning rate is not proportional to the batch size. The gradient is updated less frequent, resulting in reduced accuracy but also prone to approaching a local minimum, instead of the global minimum. This is shown in Figure 3-5, together with the effect of an adaptive learning rate. A method used sometimes with large datasets or in image recognition, for example, is to increase the batch size to what the CPU/GPU allows and then increase the number of epochs and lower the learning rate.<sup>1</sup> This typically yields in faster learning. Larger batch sizes, furthermore, tend to result in a loss in generalization performance compared to smaller batch models. This was found by LeCun et al. and Keskar et al. among others. [54, 55]

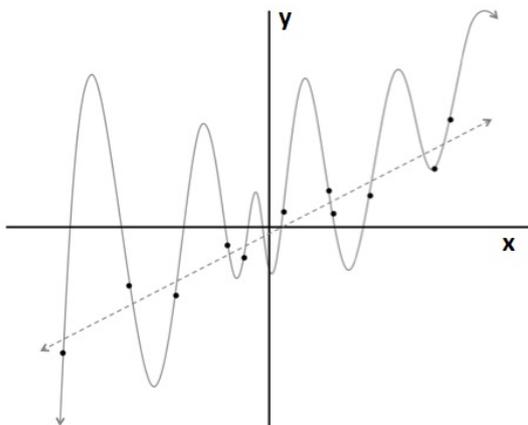


**Figure 3-5:** Batch size and learning rate impact on accuracy for image recognition [12]

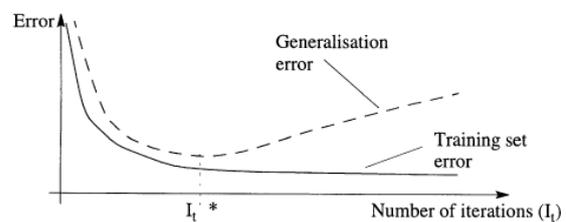
<sup>1</sup>Author's note: the shown function in Figure 3-5 for the learning rate increases the learning rate for batch size, which contradicts the earlier statement of lowering this for higher batch sizes.

## Regularization

Depending on the function of the algorithm, additional regularization techniques can be applied to the network, improving results for data samples the machine has not seen. This increases generalization of the model and decreases risk of over-fitting. The effect of regularization (or the disadvantage of over-fitting) to training data can be explained using linear regression compared to  $n$ -th order regression. Assume some data points as shown in Figure 3-6 where one would want to find a function which predicts  $y$  from  $x$ . If the  $n$ -th order polynomial were to be used (optimizing the coefficients through training) the error would be near 0; near perfect fit. While this polynomial truly describes the measurements, this function is not likely to perform well when making predictions (besides the point that training would take quite some time due to amount of unknowns). Furthermore, adding a training sample which is not on the polynomial drastically changes the optimized function. On the other hand a linear fit with a function  $y = ax + b$  (striped line) to the data points has a larger error of the loss function compared to the  $n$ -th order polynomial, but is likely to be a better model for predicting outcome. One could also assume the model to be linear with addition of some noise, so the function becomes  $y = ax + b + \text{noise}$ . This noise may be an explanation why the data is not completely linear, for example, but also contaminate the polynomial to a higher degree. Some noise added to the dataset will have a drastic effect in the unknowns in the polynomial, but barely change the linear model. In neural networks, the polynomial represents complex models, with more layers and nodes (and therefore a lot of unknowns), while the linear model represents more simple models with less layers, nodes and thus unknowns (and typically preferred). [56] Some regularization techniques are discussed below.



**Figure 3-6:** Linear (striped line) and  $n$ -th order polynomial (uninterrupted line) regression (via KDnuggets.com, modified by author)



**Figure 3-7:** Typical learning (lower curve) and testing/validation (upper curve) errors. [13]

### Early stopping

Early stopping is a method to stop the training process before it has run the full amount of set epochs, hence the name. This is typically done by setting an amount of early stopping epochs and a loss variable to evaluate. The training process is then ended if after the set amount of early stopping epochs no improvement is made. This method can reduce the required amount of epochs, but -more importantly- stops the model to a certain degree from over-fitting to the training data. An over-fitted model may have a low training error, but if provided with a test dataset (different from the training set), the test error might have increased. This is an indication of over-fitting and visualized in Figure 3-7. If the early stopping epochs are set too low the training is stopped before achieving a desired model, but if it is too high it stops the training process too late losing its effectiveness.

### Weight penalty L1/L2

Another method for regularization is weight penalty, or weight decay. This method forces connection weights values down, resulting in relative small weights. The underlying assumption is that smaller weights yield in a simpler, more stable, network than larger weights, increasing the generalization/regularization of the model. The penalty/decay tries to keep the weights close to zero, unless the gradient (during optimization) counteracts this decay. Two different loss functions are used for weight penalty: L1 and L2. Both add an additional term to the loss function. L1 adds the sum of absolute values of all weights to the cost function, while L2 adds the squared sum -hence the 2- to the cost function. Equations (3-11) and (3-12) respectively show how this is incorporated to the original, *unregularized* cost function  $C_0$ .  $\lambda$  is the regularization parameter,  $n$  the training set size and  $w_i$  the weight

$$C = C_0 + \frac{\lambda}{n} \sum_w |w_i| \quad (3-11)$$

$$C = C_0 + \frac{\lambda}{2n} \sum_w w_i^2 \quad (3-12)$$

Adding the most right term in both equations typically results in lower weights as higher weights generally increase the cost function. The regularization parameter  $\lambda$  determines how important small weights are in relation to the original cost function and is another hyper-parameter to be set. [56]

### Drop-out

Drop-out is a method which temporary leaves out some nodes (and therefore connections weights and biases) of selected input and/or hidden layers during the training process. While the combination of dropped out nodes changes after each iteration, this process continues through the whole training. This way nodes which are likely to be insignificant or redundant will get a lower weight and more important nodes a higher one. The nodes will in effect become less sensitive to the weights of other nodes, resulting in a more robust model. The nodes are not completely omitted from the network, but only left out during a training

iteration. Through this method unimportant/redundant nodes can be identified and omitted from the network. This is especially useful in large layers and/or complex models, as it will drop quite a few unknowns which have to be trained. [57]

In their paper, Srivastava et al. [57] also suggest to use a higher learning rate when using dropout to improve and speed up the training process. Furthermore, setting a maximum at the weight of 3 to 4 prevents network weights to grow very large, which is undesirable when others are not, assuming all input parameters have the same range. They also make a remark on the network size when optimizing for a dropout network. As dropping units will (likely) reduce the capacity of the network, it is advised to increase the amount of nodes in a layer to the dropout rate. Thus for dropout rate  $p$  and optimal nodes  $n$  (in a standard NN), the 'new' number of nodes in this layer should be at least  $n/p$ .

### Optimizing hyper-parameters

Combining everything mentioned in this section results in quite some hyper-parameters to optimize for. Unfortunately this is no simple optimization problem, as the solution spaces for different hyper-parameters are completely different and not always numerical (activation function for example). Also the optimizations might not be convex problems or even gradient based (e.g. the step change in amount of nodes/layers). And finally, each set of settings can only be evaluated after training of the network. This makes optimizing for hyper-parameters a very extensive process, which is typically done with grid searches. [57] Hyper-parameters have to be optimized one after another and ideally this process is repeated after all hyper-parameters are optimized in one iteration. This is, however, very time consuming and a limited amount of design iterations are typically performed. The optimization process for hyper-parameters is discussed in Chapter 6.

## 3-3 Discussion on machine learning method

In this Chapter a machine learning method is chosen from several considered options. The base of the intended models will be a neural network. This widely used algorithm has a large flexibility from a-priori knowledge, but is also capable of modeling highly complex dependencies. It works with large datasets, which could be a limitation but is also useful for ever increasing datasets.

After discussing the basics of the neural network it became clear a variety of hyper-parameters have to be determined and optimized. Some were set in this Chapter, while others have to be set through optimization which will be discussed in Chapter 6. The set hyper-parameters in this chapter are: the activation function (ELU for hidden layers, Tanh for output layer), loss function (MAE) and training or optimization function (Adam).

# Wind farm conditions

The majority of the data to be used in this research is empirical, meteorological/atmospheric data, generated by measurements performed by instruments placed on meteorological (met) masts. The electrical performance of the turbine is measured close to the generator, to reduce the impact of losses due to transport. There exist two kinds of performance measurements; those measured by a third party for Power curve verification (PCV) purposes and the Supervisory Control And Data Acquisition (SCADA) data. While results from these two are quite similar, small differences are visible due to differences in measurement device and location. To correctly pre-process the data it is important to know how, where and under which conditions these measurements are collected. There might be some correction necessary or data from certain sources might not be usable for the intended purpose at all. In this thesis project data from three different wind farms is used, being Prinses Alexia, Pen y Cymoedd (PyC) and Norrekær Enge I (NKE). This chapter discusses the site and site conditions, specifies some turbine and mast specifications and introduces the PCV in more detail as well as defining measurement (wind) sectors. More details on the different sites can be found in Appendices A, B and C for Prinses Alexia, PyC and NKE respectively. The dataset of Prinses Alexia is by far the largest and is therefore of primary interest in this thesis. Both PyC and NKE have significantly smaller datasets (over 10 times smaller) and as such are of secondary interest. In later chapters it is clearly stated which datasets are used.

### 4-1 Site descriptions

The three evaluated wind farms, Prinses Alexia, PyC and NKE are located in The Netherlands, Wales and the mainland of Denmark. Figure 4-1 shows the location of these farms.

The site conditions for Prinses Alexia and NKE are quite similar. Both have flat terrain with few objects in the vicinity. This in contrary to the hilly terrain of PyC with active forestry and can therefore be considered as complex terrain. From measurements prior to construction of the wind farms the site conditions were assessed. The average wind speed, Turbulence intensity (TI) and density for each farm are shown in Table 4-1.



**Figure 4-1:** Locations of PyC, Prinses Alexia and NKE.

**Table 4-1:** Site conditions at hub height for Prinses Alexia, PyC and NKE, respectively from [15], [21] and [22]

	Prinses Alexia	PyC	NKE
Mean wind speed	7.0 m/s	7.89 m/s	7.87 m/s
Mean density	1.251 kg/m <sup>3</sup>	1.185 kg/m <sup>3</sup>	1.255 kg/m <sup>3</sup>
Mean TI	8.9%	23.0%	4.4%

Prinses Alexia and NKE currently each have one operational met mast, while at PyC several masts are still in use (from the initial 15 located on site). From these met masts the atmospheric data is measured and averaged in 10-minute bins. Depending on the location of these masts one or more wind turbines closest to the mast are paired to the mast, so the atmospheric conditions can be linked to the turbine performance. Unfortunately not all masts in PyC can be used since not all met masts data is available at the moment of writing of this thesis and some met masts do not exist anymore. This results in only two masts, each with one paired turbine, are available to be used in this thesis. For Prinses Alexia only turbines R01 and R02 are paired to the mast, for PyC the turbines T82 and T83 are used and for NKE only turbines T04 and T05 are paired to the mast. The location of these turbines and the paired mast is shown later in this chapter. The reason for this limited selection of turbines is because the mast has to be sufficiently close to the turbine for the measured conditions at the mast to be assumed (more or less) equal, possibly through some site calibration (explained later in this chapter). If turbines would be too far from the mast not only would the flow likely be disturbed, by wake for example, but also the 10-minute averaged bins (measured at the mast and performance of the turbine) are near impossible to link to each other on the timestamps.

## Turbines and measurement equipment specifics

To correctly assess the measured parameters, the measurement equipment and their altitude is of interest. For all the paired met masts mentioned the relevant measurement instruments and altitudes at which these are placed are shown in Table 4-2. For each met mast the paired turbines are shown. For PyC the instruments are specified for both T82 and T83 together as the masts are the same. The turbine hub height is shown for reference.

**Table 4-2:** Measurement equipment altitudes located in masts linked to turbines, respectively via [15], [23] and NKE dataset.

Wind farm	Prinses Alexia	PyC	NKE
Turbines	R01 & R02	T82 & T83	T04 & T05
Hub height	98.0m	89.5m	80.0m
Cup anemometer 1	98.0m	89.5m	80.0m
Cup anemometer 2	98.0m	85.225m	78.0m
Cup anemometer 3	96.0m	62.5m	57.0m
Cup anemometer 4	46.4m	35.5m	33.0m
Ultrasonic 3D anemometer	-	82.509m	-
Sonic spinner anemometer	-	-	76.0m
Wind vane 1	96.1m	85.225m	78.0m
Wind vane 2	46.3m	31.5m	57.0m
Wind vane 3	-	-	33.0m
Pressure sensor	95.8m	83.3m	76.0m
Temperature/humidity sensor	96.3m	83.3m	78.0m

As can be seen in the table certain parameters are measured in all wind farms, like wind speed for at least three altitudes, including at hub height, wind direction at two altitudes and pressure, temperature and relative humidity. PyC also has a 3D ultrasonic anemometer and can therefore determine the wind speed and TI in all three directions. The turbine type in each relevant wind turbine in each wind park is stated in Table 4-3, together with some relevant turbine specifics.

**Table 4-3:** Turbine specifications

Wind farm	Prinses Alexia	PyC	NKE
Turbine numbers	R01 & R02	T82 & T83	T04 & T05
Turbine type	Senvion 3.4M104	Siemens SWT-3.0-108	Siemens SWT-2.3-93
Hub height	98.0m	89.5m	80.0m
Rotor diameter	104.0m	108.0m	93.0m
Rated power	3370 kW	3000 kW	2300 kW
Rated wind speed	13.5 m/s	12.5 m/s	14.0 m/s
Cut-in wind speed	3.5 m/s	4.0 m/s	4.0 m/s
Cut-out wind speed	25.0 m/s	25.0 m/s	25.0 m/s

## 4-2 Site calibration and power curve verification

Since the location of the met mast and (paired) turbines is different, the site conditions might not be exactly the same for each time bin, especially for complex terrain. Due to the surroundings the perceived wind speed can be different than the wind speed measured at the mast, for example. For this reason site calibration is sometimes performed prior to construction of the wind farm. This is primarily done for complex terrain and performed with a combination of the permanent met mast and temporary calibration mast, which is later replaced by a turbine. For a period of time the wind speed is measured at both locations, by a third party to obtain impartiality measurements, and a (linear) relation is set up to translate measured wind speed at the met mast to the location where the turbine is constructed, replacing the calibration mast. Site calibration is only performed in PyC and a previous Master thesis established the linear relations for (among others) T82 and T83, which are shown below in Table 4-4. Here  $U_{ref}$  is the wind speed as measured at the mast. No site calibration is performed for Prinses Alexia and NKE since the terrain was flat and without significant obstacles.

**Table 4-4:** Wind velocity calibration relations [24]

Turbine	Calibrated hub height velocity $U_{hub}$
T82 / UPYL05	$0.98 \cdot U_{ref} - 0.11$
T83 / UPYL06	$0.97 \cdot U_{ref} + 0.13$

## 4-3 Power curve verification

Besides the site calibration, the third party usually also performs the power curve verification, to assess whether the turbine performs as warranted by the turbine manufacturer. In the contract between farm operator and turbine manufacturer a margin is stated, and if the turbine performs less than this margin the operator has to be compensated by the turbine manufacturer. Under-performance of the turbine can therefore have serious financial consequences.

For performing the PCV the wind speed is measured at a reference mast (paired to a turbine/calibration mast) and if necessary calibrated to turbine hub height from site calibration analysis as discussed in the previous section. The turbine performance is also measured by the third party, using their own equipment. Over a period of time (in terms of months) the performance is measured and the total measured energy production is compared to the warranted energy production (by using the manufacturer's power curve and wind speed) for the same period, providing a way to assess the turbine performance. Different instruments than the ones for SCADA data are used to obtain impartial results. Since the third party uses different equipment than the operator<sup>1</sup>, there could be a difference or offset in results. Whether this offset is present and in what form can typically only be determined empirically, since the third party usually doesn't share their instrument specification (only certifications). At the writing of this thesis PCV results for Prinses Alexia are available for 2.5 years, for

<sup>1</sup>The operator uses their own equipment for SCADA

PyC for about 8 months and none for NKE. These results for Prinses Alexia and PyC are used in this thesis, while for NKE the measurements from the met mast are linked to the SCADA data, as is elaborated in Appendix C.

## 4-4 Measurement sectors

Not all (binned) wind speed directions can be considered as undisturbed upwind flow. Some wind direction bins provide for wake upwind conditions, while some yield in different upwind conditions between the met mast and the paired turbine. For these reasons some measurement sectors are defined:

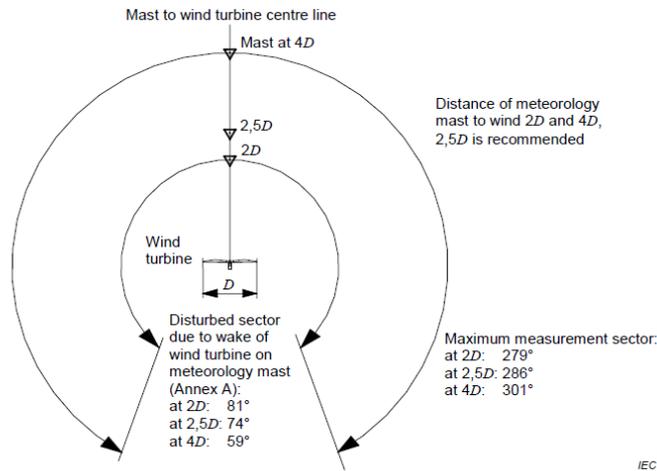
- "Free sector" - the flow is not disturbed by wake or objects in the vicinity.
- "Wake sector" - the flow is disturbed by wake from nearby turbines. No differentiation is made between the distance the wake has transversed.
- Non-similar sector - the conditions at met mast and turbine are not similar, e.g. one experiences wake. These sectors are counted as blocked sectors, discussed next, as they are not usable.
- Blocked sector - A sufficiently large object is in close perimeter of the met mast and/or turbine or when the mast is within certain distance of the paired turbine (explained in a minute). Near-wake is also included in the blocked sector.

For the free sector the International Electrotechnical Commission (IEC) defined a valid sector, which is also used for the PCV: Eq. (4-1), which indicates disturbed sectors  $\alpha$  behind disturbances.  $D_n$  and  $L_n$  respectively represent the (equivalent) diameter of and distance to the nearby turbine or obstacle. Relative distances of  $L_n/D_n \geq 20$  are considered having no influence on the met mast or turbine. [14]

$$\alpha = 1.3 \arctan \left( \frac{2.5D_n}{L_n} + 0.15 \right) + 10 \quad (4-1)$$

The wake sector is defined where both the met mast and turbine experience wake from other turbines for distances  $L_n/D_n \leq 20$ , but further away than  $4D$ , as illustrated in Figure 4-2. [14] On a more practical note, wind turbines are almost never placed closer to each other than  $4D$ , so a turbine is never 'blocked' by another, it only experiences its wake. No distinction is made between the strength of the wake, as this is a full topic on its own. Furthermore, for some wind directions the perceived wind by the met mast is different from wind as perceived at the paired turbine since only one of them is in a wake/blocked flow, which causes non-similar conditions and can therefore not be used. Valid sectors for Prinses Alexia are obtained from the PCV report [15], for PyC these are determined from data in the PCV report [16] and for NKE from the mast specifications documentation [17]. For each wind farm the wind sectors are deduced and shown in Table 4-5.

The different wind sectors are shown in Figures 4-3 and 4-4 for R01 and R02 in Prinses Alexia, Figures 4-5 and 4-6 for T82 and T83 in PyC and Figure 4-7 for both T04 and T05 in NKE.

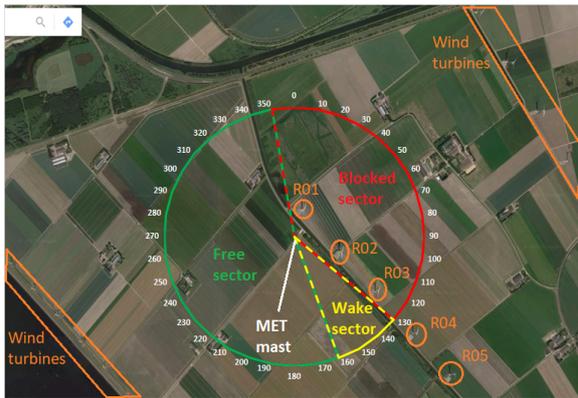


**Figure 4-2:** IEC standard on location and valid measurement sector for site calibration [14]

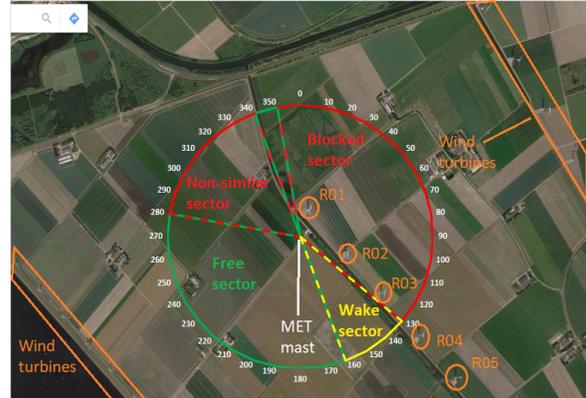
**Table 4-5:** Turbine wind sector ranges

	Free sector	Wake sector	Blocked sector
Prinses Alexia			
R01	$163^\circ - 348^\circ$	$130^\circ - 163^\circ$	$348^\circ - 130^\circ$
R02	$164^\circ - 281^\circ$	$130^\circ - 164^\circ$	$281^\circ - 343^\circ$
	$343^\circ - 348^\circ$	-	$348^\circ - 130^\circ$
Pen y Cymoedd			
T82	$220^\circ - 290^\circ$	$290^\circ - 40^\circ$	$40^\circ - 100^\circ$
	-	$100^\circ - 220^\circ$	-
T83	$218.7^\circ - 278.7^\circ$	$280^\circ - 0^\circ$	$0^\circ - 100^\circ$
	-	$100^\circ - 218.7^\circ$	-
Norrekær Enge I			
T04	$101^\circ - 229^\circ$	$85^\circ - 101^\circ$	$21^\circ - 85^\circ$
	$318^\circ - 21^\circ$	$229^\circ - 244^\circ$	$244^\circ - 318^\circ$
T05	$101^\circ - 229^\circ$	$85^\circ - 101^\circ$	$21^\circ - 85^\circ$
	$318^\circ - 21^\circ$	$229^\circ - 244^\circ$	$244^\circ - 318^\circ$

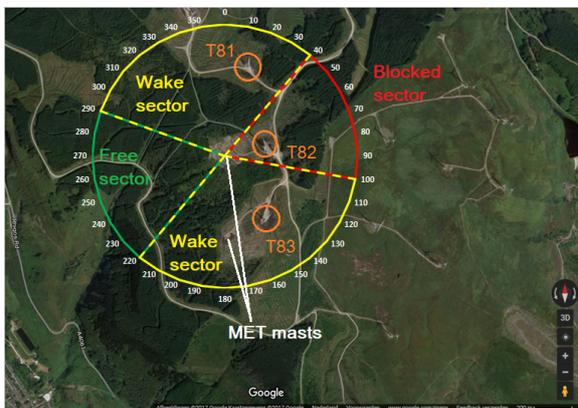
Note that in all figures, the wind direction bins are rounded to  $10^\circ$  bins. The latter figure shows the sectors where the mast experiences upwind wake from other turbines, specified in the sectors. As the turbines in NKE are placed along a line, these sectors are taken to be wake sectors or blocked if the distance from turbine to mast is too small. As stated, distances of over  $4D$  are considered wake, closer than  $4D$  (near wake) is considered blocked. Data from the free sector is the only data assessed as no 'artificial' contamination of the data due to wake or blocked flow is present in the data and the atmospheric influence can be singled out.



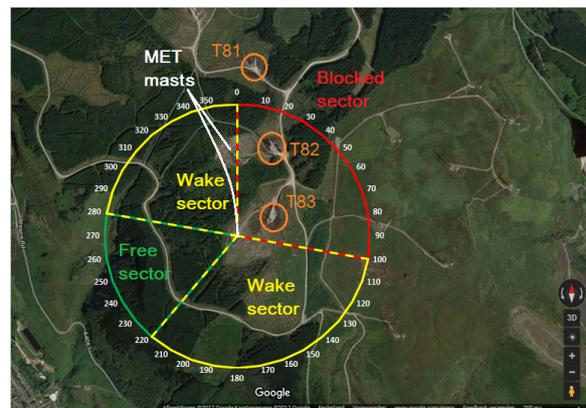
**Figure 4-3:** Wind sectors for turbine R01, in Prinses Alexia, and paired met mast.<sup>2</sup>



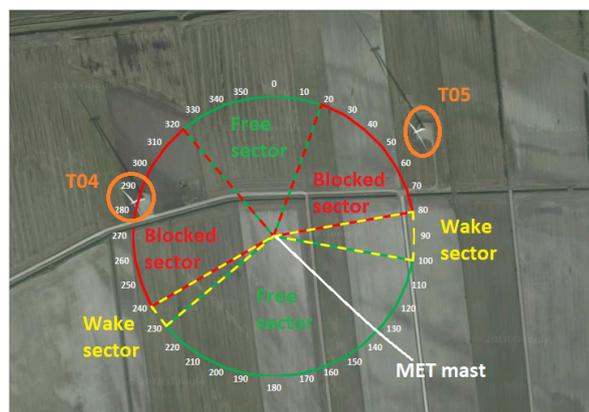
**Figure 4-4:** Wind sectors for turbine R02, in Prinses Alexia, and paired met mast.<sup>2</sup>



**Figure 4-5:** Wind sectors for turbine T82, in PyC, and paired met mast (at the center of the wind rose).<sup>2</sup>



**Figure 4-6:** Wind sectors for turbine T83, in PyC, and paired met mast (at the center of the wind rose).<sup>2</sup>



**Figure 4-7:** Wind sectors for turbines T04 and T05, in NKE, and paired met mast (at the center of the wind rose).<sup>2</sup>

<sup>2</sup>Map via Google maps

## 4-5 Discussion on measurement conditions

This Chapter introduced the wind farms from which the dataset, used in this thesis, are obtained. The locations, site conditions and measurement devices from which the empirical data is obtained are discussed, along with some turbine characteristics. Power curve verification is introduced, which will be important in one of the models. Finally measurement sectors in each wind farm were introduced, splitting wind direction bins in the free, wake and blocked (upwind) stream. While this decreases the usable dataset sizes, it improves the accuracy of the data samples in the subset and allows for investigating undisturbed flow. In the next Chapter it will become clear what the impact on the usable dataset is.

# Pre-processing

While the 10-minute averaged measurements can lead up to quite a large dataset, not all data samples make sense and can be used in future analyses. There can be a multitude of reasons for this, such as malfunctions or errors, damages or environmental issues. Since contaminated data or noise is highly undesirable, such data points have to be filtered out. Including such points will result in a less accurate model as the model will be trained for these contaminations. An advantage of using data from a relative new wind farm, located in a temperate climate, is the wind turbines are very likely to be in good condition, so (significant) blade degradation is not considered. This chapter discusses filtering (techniques) applied to the raw datasets and elaborate on why they improve the dataset. Finally an initial selection is made of parameters which are likely to have an influence on performance. These parameters are finally evaluated as input for the neural network model.

### 5-1 Data filtering

To obtain a dataset of usable data points, points deemed incorrect are omitted from the raw dataset. There is a range of situations for which to reject data points, but there is not one truth which samples to reject. The International Electrotechnical Commission (IEC) standard advises data rejection on the following criteria [14]:

- Wind speed outside of operating range
- Wind turbine fault condition
- Wind turbine normal shut-down or maintenance operating mode
- Failure or degradation (icing for example) of measurement equipment
- Non free-stream wind directions
- Invalid site calibration wind directions

Extending on this advice, common practice is to initially classify data points in types, shown in Table 5-1. From this table, a domain of valid data points is sketched where only points of type 1 are left in the dataset, types 2 to 5 are omitted. While for type 4 and 5 there is a clear benchmark, this is not so clear with points 2 and 3. How to handle this ambiguity is treated later in this section. This classification is also used by Kusiak et al. and Schlechtingen et al. [20, 51] In the following sections different methods of filtering and their purposes are discussed. Altogether the following nine filters are applied to the data, which are shown in Table 5-2 and discussed next.

**Table 5-1:** Classification of data points [25]

Type	Data description
1	Data points following the pattern of the turbine's power curve
2	Data points with high wind speed and low power values
3	Data points with low wind speed and high power values
4	Data points with negative wind speed values
5	Data points with negative power values

**Table 5-2:** Applied filters and their order

Filter order	Filter description
1	Farm or turbine alarms
2	Missing data points
3	Error flags
4	Negative or zero power values
5	Velocity limit
6	Minimum temperature
7	Wind direction sector
8	Time boundaries
9	Abnormal data points

## Alarm filtering

In the Supervisory Control And Data Acquisition (SCADA) database, turbine alarm periods are stored. This means for some periods of time the power of a wind turbine or the complete wind farm is truncated due to various reasons, for example maintenance or grid failure, and therefore did not perform as it should. These data points are completely omitted from the dataset. Periods of turbine curtailment are considered under alarms, as the turbine does not perform as it should be for given conditions. Curtailment of a turbine means the turbine is manually switched to lower performance, for energy trading/price purposes, noise reduction or icing for example.

## Missing data points

Due to various reasons empty values for one or more parameters in data points might occur. These obviously contaminate or even disrupt the dataset. Even if the parameter itself is not used, there is a possibility the cause of the empty value has an influence on other parameters. For this reason data points with one or more empty/missing values are completely omitted from the dataset.

## Error flags

Slightly similar to alarms, but from a different origin are error situations. These situations are captured by error flags, indicated in the third party and SCADA datasets. The dataset contains error flags with a value between 0 and 1, where a non-zero value represents at least one error in all 1 Hz measurements in the 10-min averaged bin. As the cause of the error is not indicated, the severity of the error is hard to assess. Error values of over 1% (6 errors in the 1 Hz measurement domain) are omitted from the dataset. This may seem a hard benchmark, but even small amounts of error can cause potential problems for the network and therefore accuracy. Errors of less than 1% are assumed to have an insignificant effect -if any- on the averaged bin and is therefore set to be the benchmark.

## Negative and zero power and wind speed filtering

Data points with zero or negative measured power values are rejected from the dataset, as well as points with negative measured wind speed. As wind speed is measured with a cup anemometer, negative wind speeds should not occur and is likely to indicate an error. Negative power is typically caused by wind speeds under cut-in wind speed, which are also to be omitted from the dataset (also discussed next). Zero power data samples are also rejected as turbine shutdown -for any reason- is not to be modeled.

## Velocity limits

Typically a wind turbine is shut down for wind speeds under cut-in and above cut-out wind speed. However, turbine performance above cut-out does not always need to be zero due to Siemens' *High wind ride through* or Enercons *System control* systems, which provide for a way to gradually, though rapidly, lower the performance of turbines for high wind speeds. These are not modeled for as it would only be convenient if all used turbines had similar systems. Turbine shutdown, including below cut-in or above cut-out, is captured in an error flag and these points are omitted from the dataset, as discussed earlier. However, after manually investigation of the dataset one data point with an extremely negative wind velocity in one of the wind speed measurements ( $-6666m/s$ ) was found. It is very likely this is caused by a incorrect data-logger conversion of the signal from the anemometer. To remain with the operational range of the wind turbine an additional filter is set up, removing wind speed measurements below cut-in and above cut out wind speeds.

## Minimum temperature benchmark

The next 'layer' of filtering is becoming less absolute than the previous ones. While failure of measurement instruments is typically quite clear, degradation of equipment is not so black-and-white. Degradation is assumed to occur when atmospheric conditions tend to influence the measurement device in such a way that it does not yield in correct results and is typically caused by icing for example. The problem lies in defining the border from correctly operating devices and degraded devices. The IEC advises only on a safe temperature of above 2°C. [14] Conditions below this benchmark temperature are assumed to cause icing in the measurement device rendering the measurement incorrect. The temperature benchmark is, however, set as 0°C as the advised 2°C is widely considered as a very safe benchmark and icing is not a known problem for the considered wind farms.

## Wind direction and time window splitting

As mentioned at the start of this section, the IEC also advises on filtering for wind direction such that only the unblocked, free stream is taken into account. For site calibration an additional advice is given, related to the meteorological mast. While the undisturbed flow is the only wind direction sector considered in this thesis work, the datasets are also split for the wake flow, since this might be of interest in the future. In the same reasoning it might be interesting to see how certain parameters and their influence on performance change over the time of day. This is, however, not of primary interest in this thesis and no time constraints are considered.

## Abnormal data filtering

This method determined whether points are within a set range from the measured mean and classifies points outside a certain range to be 'abnormal', rejecting them from the dataset. This method assumed the measurement points to Gaussian distributed for each wind speed bin. This method has been used in power curve modeling extensively. [1, 20, 25, 51]

To analyze data points, first the dataset is divided in wind speed bins with a range of 0.5 m/s. This bin range can be smaller with sufficient available data points, but in this research the used range was 0.5 m/s to keep sufficient data points in most bins to still assume a normal distribution. For each bin the unbiased mean power ( $\mu$ ) is determined using Eq. (5-1), where  $N$  is the dataset length and  $P_i$  the power for each sample. Then the standard deviation  $\sigma$  of the bin is determined using Eq. (5-2).

$$\mu = \frac{1}{N} \sum_{i=1}^N P_i \quad (5-1)$$

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (P_i - \mu)^2} \quad (5-2)$$

From these mean and standard deviation for each bin the upper and lower validity limits can be set. The Upper control limit (UCL) and Lower control limit (LCL) are determined by the researcher, using Eq. (5-3) and Eq. (5-4) respectively (modified equation via [51]).

$$\text{UCL}_i = \mu_i + n\sigma_i \quad (5-3)$$

$$\text{LCL}_i = \mu_i - n\sigma_i \quad (5-4)$$

In these equations the power limit for each bin  $i$  is set from the mean  $\mu$  and standard deviation  $\sigma_i$ . The constant  $n$  is present to influence the sensitivity of outliers. The higher the value the less outliers are filtered. Kusiak et al. have used  $n = 2 \cdot \sqrt{2}$ , to make the model less sensitive on the data variability. However, in this research  $n = 2$  is chosen, to filter out the outlying outer 5%. This yields in less data points, but also with less noise or outliers which contaminate the model.

## 5-2 Atmospheric influences

Not all relevant atmospheric influences are directly measured by the met masts or instruments on the wind turbine. Some have to be derived from other measured parameters which include the Turbulence intensity (TI), wind shear component, wind veer, density and inflow angle. These parameters and their methods for calculating these were discussed in Chapter 2.

- TI is determined using the average wind speed and wind speed standard deviation in a 10-minute averaged bin, both measured by the cup anemometer, as defined in Eq. (2-4).
- Wind shear is determined using the average wind speed at different altitudes as measured by available cup anemometers. Eq. (2-5) is used for determining the wind shear component  $\alpha$ . This translates into Eq. (5-5). For part of the dataset the calculated wind shear component  $\alpha$  is negative. While it may seem counterintuitive for the wind speed to decrease with altitude in the lower atmospheric layer, negative wind shear component is a sign of a highly stable atmospheric stability or Lower level jet-stream (LLJ). [33]

$$\alpha = \frac{\log\left(\frac{U_i}{U_{hub}}\right)}{\log\left(\frac{h_i}{h_{hub}}\right)} \quad (5-5)$$

- The wind veer is determined from the wind direction measured on two different altitudes and the gradient is linearly determined (as typically wind direction is measured at 2 altitudes) using Eq. (2-6).
- Density is determined from the measured temperature, relative humidity and pressure using Eq. (2-12).
- Since a Ultrasonic (US) anemometer is only present at Pen y Cymoedd (PyC) and a spinner anemometer at Norrekær Enge I (NKE), the inflow angle could not be calculated for every farm, which is therefore not evaluated.

## Supersaturation

A selection of the data-points in the dataset have measured humidity values of over 100%, which seems incorrect. However, this data is measured and processed by the third party and the measurement instruments are verified. While in the Prinses Alexia dataset values of over 100% are not present (just like in the dataset for NKE), they are in the PyC dataset. What would be a reason for these values of over 100%? The answer lies in supersaturation. Supersaturation is the condition, in meteorology, when the air contains more vapor than is required for saturation. This occurs if there is sufficient vapor in the air while also an absence of condensation nuclei or wettable surfaces for the vapor to condensate. Theoretically Relative humidity (RH) of 400% is possible. [58]

## Site calibration

Prinses Alexia is located on very flat terrain, just like NKE, and the met mast is close to both used turbines, which were also evaluated for Power curve verification (PCV). For these reasons no site calibration has been taken in the contract and therefore performed. 10-minute averaged measured wind speed at the met mast can be linked directly to the turbine location. PyC, however, is located on a complex terrain and execution of site calibration is defined in the contract between turbine manufacturer and farm contractor and therefore a relation between wind speed at the reference mast and specified turbines exist. The relations were earlier defined by a previous Masters thesis and are specified in Table B-4 in Appendix B. These relations are directly used to correct the wind velocity at hub height in the measurement datasets.

## 5-3 Filtered dataset evaluation

As mentioned Prinses Alexia is the main wind farm of interest due to the size of the available dataset. However, the datasets for PyC and NKE are also prepared for further data analysis. These are respectively shown in Appendix B and Appendix C. For NKE there is an additional step performed, as the data from the mast was not linked to the measured performance of the turbine and timestamps from both datasets did not match. This is also elaborated upon in Appendix C.

The raw dataset of Prinses Alexia consists out of 124.272 data samples and ranges from February 2015 to September 2017, as provided by WIND-consult GmbH (WICO). The resulting amount of omitted and remaining samples in the datasets for turbines R01 and R02 are shown in Table 5-3 and Table 5-4 for the free and wake stream respectively. For reference, the final datasets size of PyC for the free stream is 7699 samples, the final dataset of NKE 6013 samples.

**Table 5-3:** Prinses Alexia turbines lost data samples due to filtering in free stream

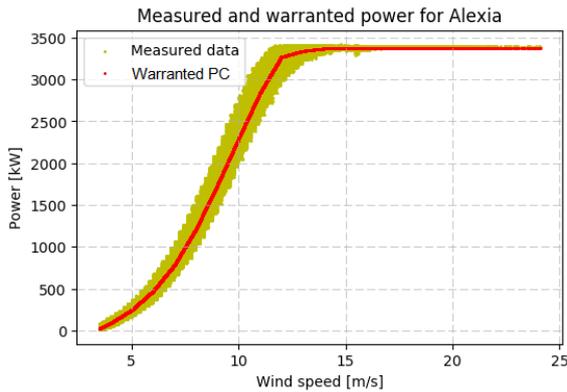
Turbine Sector	R01 290° - 40° & 180° - 220°		R02 280° - 0° & 100° - 220°	
	Remaining points	Removed points	Remaining points	Removed points
Filter method				
Combining dataset	124272	-	124272	-
1. Alarms	121917	2355	122656	1616
2. Missing points	106086	15831	120275	2381
3. Error flags	98225	7861	113388	6887
4. Non-zero power values	79604	18621	90846	22542
5. Velocity limits	79366	238	90781	65
6. Minimum temperature (0°)	77254	2112	88607	2174
7. Wind direction boundaries	50953	26301	48983	39624
8. Time boundaries	50953	0	48983	0
9. Abnormal data filter	<b>44701 (35.97%)</b>	6252	<b>42445 (34.03%)</b>	6538

**Table 5-4:** Prinses Alexia turbines lost data samples due to filtering in wake stream

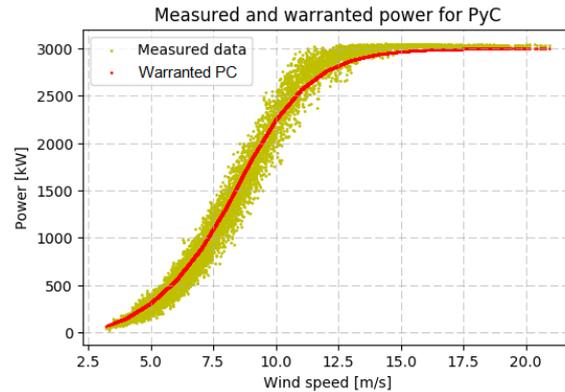
Turbine Sector	R01 163° - 348°		R02 164° - 281° & 343° - 348°	
	Remaining points	Removed points	Remaining points	Removed points
Filter method				
Combining dataset	124272	-	124272	-
1. Alarms	121917	2355	122656	1616
2. Missing points	106086	15831	120275	2381
3. Error flags	98225	7861	113388	6887
4. Non-zero power values	79604	18621	90846	22542
5. Velocity limits	79366	238	90781	65
6. Minimum temperature (0°)	77254	2112	88607	2174
7. Wind direction boundaries	15412	61842	17783	70824
8. Time boundaries	15412	0	17783	0
9. Abnormal data filter	<b>13811 (11.11%)</b>	1601	<b>15915 (12.81%)</b>	1868

## 5-4 (Filtered) measured performance

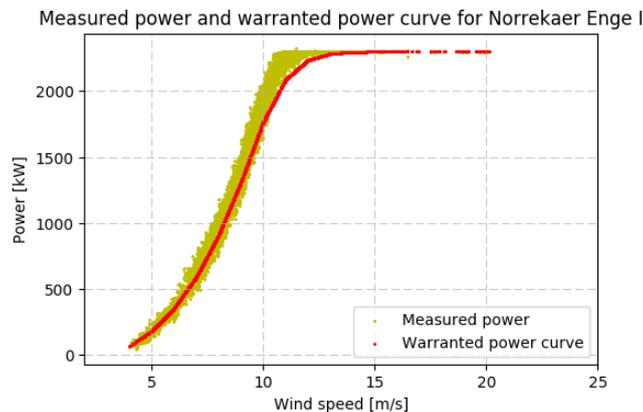
To get an idea how the measured performance stands in comparison to the warranted power curve, all filtered points are plotted, measured performance versus the wind speed, together with the Warranted power curve (WPC). Only the free stream sectors are plotted here, because these are by far the largest datasets and these are the primary interest in this thesis. The plots for Prinses Alexia, PyC and NKE are shown in Figure 5-1 to Figure 5-3 respectively. While the WPC represents the measured power on average well, it is off at the knee of the power curve for PyC, but more drastically for NKE. Especially for NKE the WPC seems to be a serious under-prediction. Furthermore for both Prinses Alexia and PyC the WPC is slightly less than measured power for variable pitch/constant power. Both of these trends are sure to be notices in trained models.



**Figure 5-1:** Measured performance in the free stream sector and warranted power curve for Prinses Alexia



**Figure 5-2:** Measured performance in the free stream sector and warranted power curve for PyC



**Figure 5-3:** Measured performance in the free stream sector and warranted power curve for NKE

## 5-5 Network input and output

Of the over 180 parameters present in the original database, obtained for Prinses Alexia, a limited amount is usable. Some are not usable as they are signals from a logger, while others provide information about the 10-minute averaged bin (minimum, maximum, standard deviation) for example. During pre-processing most of these are removed and a selection is made for parameters to be used as input to the neural network.

Selecting the right (or wrong) parameters as input for a neural network has some consequences. While it is no disaster if a useless parameter is selected, it adds unnecessary complexity to the model, resulting in a less than optimal model. The most important reasons for selecting certain input variables are stated below. As the output of a network is preferably between -1 and 1 (or 0 and 1), a normalization scheme for the output parameter has to be set which is done after the input variable selection.

## Input parameters

To choose the input parameters for the neural network, there are some -mainly practical- limitation. In theory the amount of input parameters is unlimited, but this is of course not viable. The most important limitations are:

- Input parameters should coincide. So the input parameters for one (input) farm should be the same as the farm for which the model is then tested. These should also have the same meta-conditions, such as relative altitude and measurement direction, relevant to the parameter.
- The parameters should (preferable) be independent. Density for example, is calculated from three measured parameters and such a relation should be found in a sufficient complex neural network. Adding a dependent variable will complicate the training process unnecessarily.
- The dataset size should be sufficiently large to train a model with the set amount of input parameters. Adding more input parameters results in a more complex network, with more unknowns and therefore requires a larger dataset.

Based on above limitations, the following parameters are chosen as input for the neural network:

- $U_{hub}$  - Hub height average wind speed [m/s]
- $T_{amb}$  - Ambient temperature [°C]
- $RH$  - Relative humidity [%]
- $p_{amb}$  - Ambient pressure [hPa]
- $I_x$  - Longitudinal turbulence intensity [-]
- $\alpha$  - Wind shear component [-]
- $\varphi$  - Wind veer [°/m]

## Output parameter and normalization

While the output of a neural network does not necessarily need to be normalized, doing this will reduce range of the output and therefore reduce the magnitude of connection weights, resulting in a more stable network. A normalized output, instead of the absolute power for example, would be more generally applicable and not just for one type of turbine in a single wind farm. For this reason the normalized difference from the warranted power curve was chosen as normalization scheme (Eq. (5-6)). This scheme has a domain between -1 and 1, and incorporates the shape of the warranted power curve. The output basically becomes the refinement, or adjustment, to the warranted power curve for certain external, atmospheric conditions. An additional -assumed- advantage of this method over 'direct' normalization is the range of the model output is fairly limited to the proximity of the WPC and therefore

flattens out the modeled power (refinement). This potentially yields in less over-fitting on the (normalized) measured power. Both of these normalization schemes are compared in Appendix E, Section E-3 for both the model for a single wind farm.

$$y_{train} = \frac{P_{measured} - P_{warranted}}{P_{rated}} = \frac{P_{measured}}{P_{rated}} - \frac{P_{warranted}}{P_{rated}} \quad (5-6)$$

$$y_{train} \in [-1, 1]$$

## 5-6 Discussion on data pre-processing

In this Chapter various filtering methods were shown, which were applied to the raw datasets of all wind farms. The datasets were filtered for alarms and errors, empty data samples, samples with negative or zero power and for measured velocity beyond the power curve range, which are all fairly straight forward. For the minimum temperature and abnormal data samples, the user/researcher has to set some benchmarks to determine which points to omit. Finally the dataset is split into the three wind direction sectors; free, wake and blocked. From which only the free stream is used in this thesis. The combined filters applied to the raw dataset resulted in a loss of about 65%-70% of data for each wind farm, drastically reducing the amount of available data samples.

Finally the input and output of the neural network are discussed. The input parameters were selected from all available input parameters. The model input is thereafter set to consist out of: wind speed, ambient temperature, relative humidity, ambient pressure, longitudinal turbulence intensity, wind shear component and wind veer. For the output a normalization scheme was determined, to improve general use of the intended models. The output parameter will be the difference from the warranted power curve, normalized to rated power. By doing this information about the turbine's power curve shape is retained, while still having a normalized output parameter.

# Setting up a neural network based model

With the machine learning method set to a Neural network (NN), the datasets available and pre-processed, the model can be constructed and trained. This is done using existing software packages as a base, as designing a neural network software from scratch is beyond the scope of this project and likely beyond the capabilities of the author. In this chapter the software and packages are briefly specified first, followed by a global overview of the model. Then the selection of the hyper-parameters is discussed, which was elaborated upon in Section 3-2.

## 6-1 Neural network software

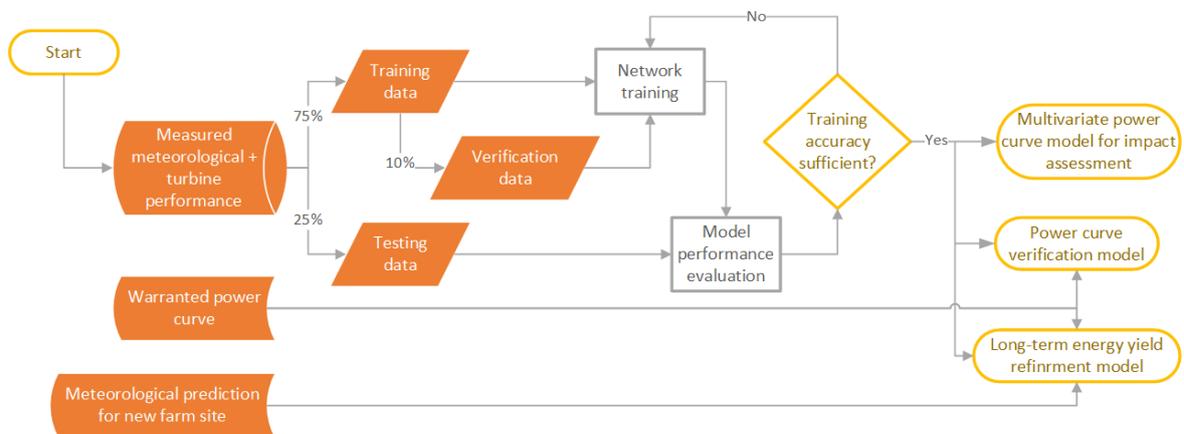
The programming language used for this thesis is Python [59], which is chosen above Matlab for example, for two reasons. First Python is open-source and a lot of packages are freely available on-line. The second reason is that the author is more proficient with Python than Matlab. A small note: Python is comparable with *R* [60], used by the analytics team within Nuon/Vattenfall so they can easily adapt written scripts, should the result of this thesis be of interest to them.

For neural network packages, a lot are available and were assessed. Since the analogy of a neural network is more or less the same for all packages, user-friendliness, community support, efficiency (in terms of computational power) and flexibility were compared. This resulted in the choice falling on *Keras* [27]. *Keras* is a neural network Application programming interface (API) for Python, running on top of *TensorFlow*, *CNTK* and *Theano*. *Keras* is considered user-friendly, modular and allows for setting up a basic network relatively fast. For this project *TensorFlow* is used, which is an open source library developed by Google, specializing in machine and deep learning applications.

## 6-2 Model overview

Figure 6-1 shows the flowchart of the intended model. The available data is first split in training and testing data (75%/25%). From the training data 10% is used as validation data *during* the training process, this subset will be randomly chosen after each epoch. Optimization of the hyper-parameters is done in k-fold, during which the training process is performed  $n$  times for the same dataset, after which the results are averaged. This will reduce dependency of the solution on stochastic behavior of the training function.

After each training process the model will be tested with the testing dataset and evaluated for performance. How this is evaluated is elaborated upon for each hyper-parameter optimization, described in later sections.



**Figure 6-1:** Flowchart of all intended models, each indicated in green.

Three distinct models (and their names) can be deduced from the overview:

1. A generalizing model to refine long-term energy yield estimations. This will henceforth be called the *Business Case Refinement (BCR)* model, as it refines the prediction made for wind farm tender business cases.
2. A model to continue power curve verification after the verification period through Supervisory Control And Data Acquisition (SCADA) data. This will be called the *Power curve verification (PCV)* model, for obvious reasons.
3. A model to single out influence of individual parameters on turbine performance. This model will be called the *Performance Correlation Assessment (PCA)* model, as it assesses the correlation of individual parameters to the turbine power.

The latter two are quite similar, as these are trained with data from a single wind farm/turbine and lack a certain amount of generalization. Below the general approach of optimizing the hyper-parameters is described, but only showed for one model: the BCR model. This one is shown since the most hyper-parameters have to be set for this model, the other two will not have all generalization parameters. The optimization for the models on a single wind farm, the PCV and PCA models, are the same and the hyper-parameter searches is shown in Appendix F.

## 6-3 Hyper-parameters

In Chapter 3 the basics of a neural network were explained. A variety of hyper-parameters were mentioned, which have to be optimized for setting up a good neural network. This optimizing/tweaking of (a lot of) hyper-parameters is what makes setting up a neural network quite extensive. [57] While some people claim rules of thumb exist for such hyper-parameters and functions, the best settings greatly depend on the problem itself. Therefore no rules of thumb are used for setting the hyper-parameters. While the activation, loss and optimization functions are reasoned and set, the network layout, batch size and generalization parameters are to be determined by means of grid searches. After a full search of all hyper-parameters, a subsequent design iteration could be performed. This was not done, since it is assumed to provide for a minor improvement in results, while an extensive period of grid searches needs to be performed. So only one design iteration is performed in optimization of the hyper-parameters.

### Evaluation method

In evaluation of the model accuracy two aspects are compared for each model setting; the Mean absolute error (MAE) and the  $R^2$ -score between the modeled performance and the measured performance. The MAE has been discussed earlier in Section 3-2 and again shown in Eq. (6-1). The  $R^2$ -score indicates how well the modeled samples compare to the measured samples. The  $R^2$ -score ranges from 0 to 1, where high scores indicate modeled output is, per sample, close to the (training) output. This score is calculated using Eq. (6-2), using Eq. (6-3) and Eq. (6-4) for the residual and total 'sum of squared' ( $SS_{res}$  and  $SS_{tot}$ ) respectively. [61] Here  $y_i$  is the measured output,  $\hat{y}_i$  the modeled output and  $N$  the number of samples.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (6-1)$$

$$R^2 \equiv 1 - \frac{SS_{res}}{SS_{tot}} \quad (6-2)$$

$$SS_{res} = \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (6-3)$$

$$SS_{tot} = \sum_{i=1}^N \left( y_i - \frac{1}{N} \sum_{i=1}^N y_i \right)^2 \quad (6-4)$$

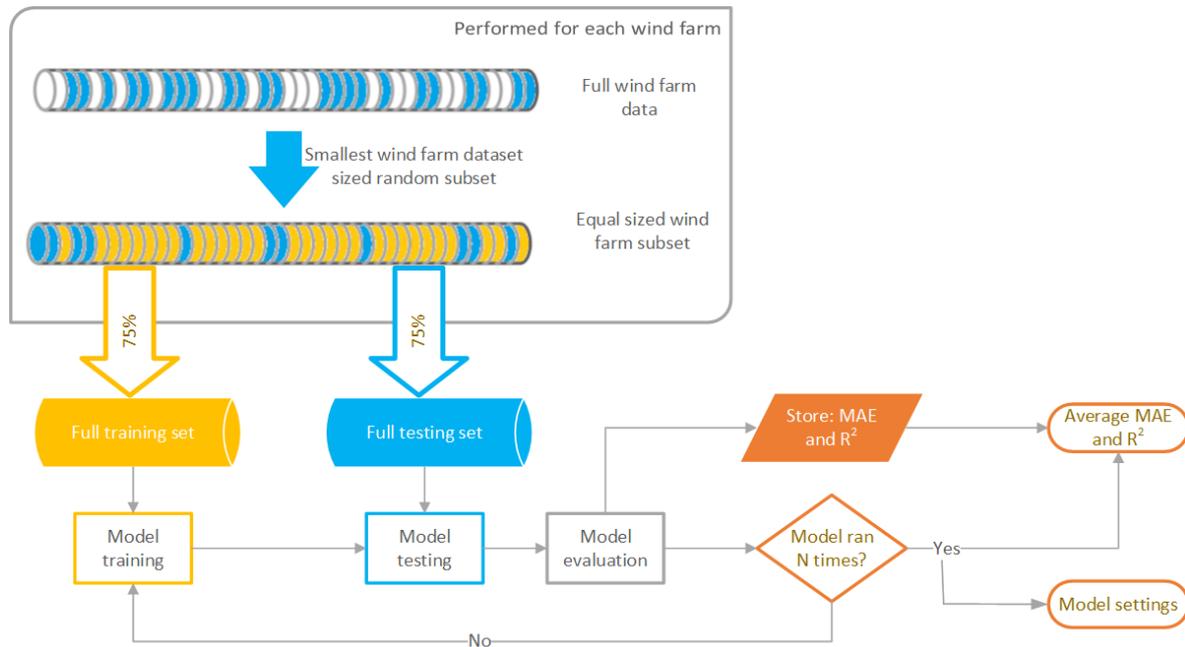
There is a distinction in training and test data between the stated models. The PCV and PCA models use a dataset from the same wind farm, where 75% is used as training data and 25% as test data. These models should be trained for individual wind farms or turbines, without regularization. The settings with the lowest error and  $R^2$ -score closest to 1 yields in the best model. Because the dataset size of Pen y Cymoedd (PyC) and Norrekær Enge I (NKE) are quite small (respectively 7367 and 6013 samples in total) these are deemed too small to accurately and consequently train the network. Therefore only Prinses Alexia (total dataset size 86,883) is used to train these two models. The BCR model, however, uses a combined dataset of PyC, Prinses Alexia and NKE in which each wind farm contributes equally to the total dataset to prevent one farm being dominant in training data. 75% of the (equally sized) dataset of each farm is used for training and 25% for testing. Modeled results for this test set are evaluated for each wind farm individually. The best settings are those for which the MAE is lowest and  $R^2$ -score closest to 1, but also were both are closest to each other for all wind farms to prevent the model to be dominant for a single wind farm. Root mean squared error (RMSE) (Eq. (6-5)) is also considered in evaluating model performance, as a secondary parameter besides MAE.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (6-5)$$

In the optimization function, Adaptive moment estimation (Adam), there is some randomness built in, as it is a (partly) stochastic solver. Having sufficient data samples removes the dependency of the solution on this randomness. Since the combined training dataset is considered not sufficient to omit this dependency on the randomness, 5-fold cross validation is used. This means that the training process is run 5 times with the same settings and the average values for MAE and  $R^2$ -score is taken. These resulting averaged values are the results shown in this chapter. Note that the training and test subsets are kept the same for each training and testing run. An overview of this process is shown in Figure 6-2, where the upper container ("*Performed for each wind farm*") indicates each wind farm contributes an equal sized training and test subset to the full training and test dataset. By doing this procedure the only randomness occurs in the optimizing function, which yields in the best method to compare model settings. The choice of 5-fold, instead of a larger amount, is set because of time constraints. 5-fold resulted in just over a week of grid searches.

## Architecture

Recall that the architecture of the network means the amount of layers and nodes per (individual) layer. Larger amounts of nodes and/or layers does not necessarily result in a better model as considerably more unknowns have to be solved, thus training becomes more complex. Additionally there is the risk a too complex model, over-fitting on the training data, which results in a model with less accurate predictive capabilities as was discussed in Section 3-2. Too few, however, might result in a model not being able to model the desired complexity level. Therefore a range of lay-outs is considered and evaluated. The maximum amount of hidden layers was set at 3 and the maximum amount of nodes per layer was set at 16 at max (little more than twice the amount of input parameters). The maximum amount of internal unknowns (weights and biases) would then be:  $7 \cdot 16 + 16 \cdot 16 + 16 \cdot 16 + 16 \cdot 1 = 640$  weights



**Figure 6-2:** 5-fold cross validation. The dataset remains the same for each training process, but the training process itself is evaluated 5 times for each setting, averaging the results per setting.

and  $16 + 16 + 16 + 1 = 49$  biases, so 689 unknowns. Using 33%-33%-33% Prinses Alexia, PyC and NKE datasets of equal length, the total dataset will be 18,039 samples, resulting in just about 13,530 training samples or about 19.5 times more samples than unknowns. Especially for smaller networks this should be sufficient to find the global solution for the BCR model. For the other two models, the dataset of Prinses Alexia is used, consisting out of 86,883 training samples or 94 times more samples than unknowns, which is considered large enough for accurate training and finding the global minimum. If the other two farms were used, however, it is likely a different local minimum are found during each training run yielding in unreliable training of the model.

The grid search is performed for base settings which can be found in Table 6-1, no generalization methods are added yet. The batch size and maximum number of epochs is chosen such that the training does not take too long and full grid search would take 5 days at most. Earlier, in Chapter 3, the activation and optimization functions were reasoned and set.

Since the grid search was performed for over 800 cases, only to be presentable in a table, this grid search is not shown in this report. The resulting optimal network layout, however, is; [14-8-12-1] nodes in sequential layers. The MAE, RMSE and R<sup>2</sup>-score for the warranted and modeled performance compared to measured performance are shown in Table 6-2. When dropout is introduced to the network, the nodes per layer will increase as was mentioned in Chapter 3.

**Table 6-1:** Grid search hyper-parameters initial settings

Hyper-parameter	Value
Max. epochs	400
Batch size	512
Test part of full dataset	25%
Hidden layer activation functions	Exponential linear unit (ELU)
Output layer activation function	Tanh
Optimization function	Adam
Early stopping epochs	0.0 (no early stopping)
Drop-out rate	0.0 (no dropout)
Regularization penalty	0.0 (no penalty)
Training set	33.3% Prinses Alexia 33.3% PyC 33.3% NKE

**Table 6-2:** Model performance after hyper-parameter grid searches. The test set contains the equally sized test sets for all three wind farms combined.

'Standard' network layers lay-out: [14-8-12-1]			
	MAE [kW]	RMSE [kW]	R <sup>2</sup> -score
Warranted power curve	94.04	121.43	0.9702
Modeled	62.88	80.77	0.9868

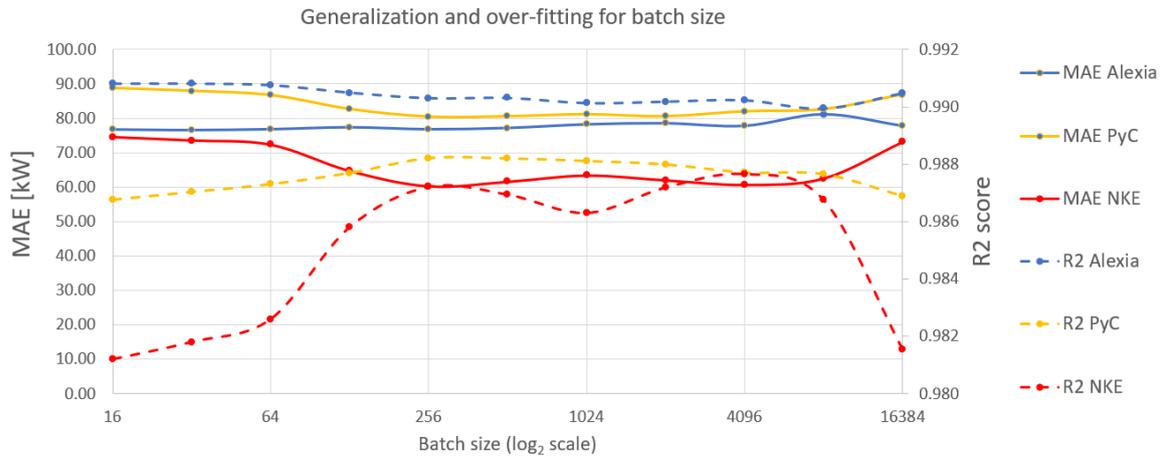
## Epochs and batch size

Recall that batch size means how many data samples, out of the full training dataset, are propagated through the network in one iteration, while the number of epochs is the amount of times the *full* training set is propagated through the network. In other words:

- Samples per iteration: Batch size
- Iterations per epoch: Training set size / batch size
- Samples through 1 epoch: Iterations · batch size per iteration

The total number of epochs is now set 10 times larger, at 4000. When early stopping is implemented this amount is not likely to be reached as the solution will be considered converged before reaching the 4000 epochs. For now, however, this number is set to determine the batch size, which is found by means of a grid search. Due to memory usage batch sizes are set at multitudes of the power of 2, starting with 16 samples up to 8192 samples per batch. Validation is done for each wind farm individually and again the size and division randomness of the full datasets are kept equal and constant for all wind farm subsets. The resulting performance change for each wind farm is shown in Figure 6-3.

While for Prinses Alexia there is minor change in both MAE and R<sup>2</sup>-score, the best MAE and R<sup>2</sup>-score for both PyC and NKE occurs at a batch size of 256. This is therefore the batch size chosen for next grid searches and the final network.



**Figure 6-3:** Batch size grid search test dataset results, evaluated per farm individually.

## Regularization

Next the hyper-parameters for regularization, which are the drop-out rate, regularization parameter and early stopping epoch amount, were varied to find the optimal values. In the same manner as for the batch size, the individual wind farm test sets were evaluated and plotted against each varying hyper-parameter range. The initial network settings, shown in Table 6-1, are adjusted with a batch size of 256 and when a regularization parameter is found, this is updated in the network settings for the next grid search.

### Early stopping epochs

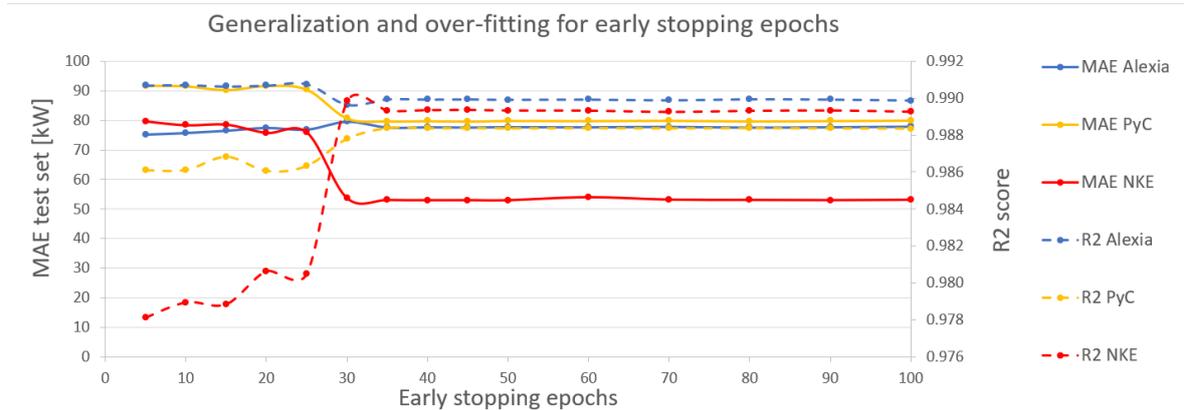
As explained in Section 3-2 the early stopping method stops the training process when after a number of epochs no improvement has been achieved, preventing some over-fitting of the network. The tested range of early stopping epochs was set at 5 to 100 with an interval of 5 epochs up to a value of 40 and 10 for higher early stopping amounts. The resulting MAE and  $R^2$ -score are plotted in Figure 6-4.

As can be seen the amount of early stopping epochs converges after 35 epochs for all wind farms. to be slightly safer, the early stopping amount is set at 40. With this amount of early stopping epochs set, the total amount of training epochs required to train the network is on average about 1150 and thus far below the set maximum amount of epochs (4000).

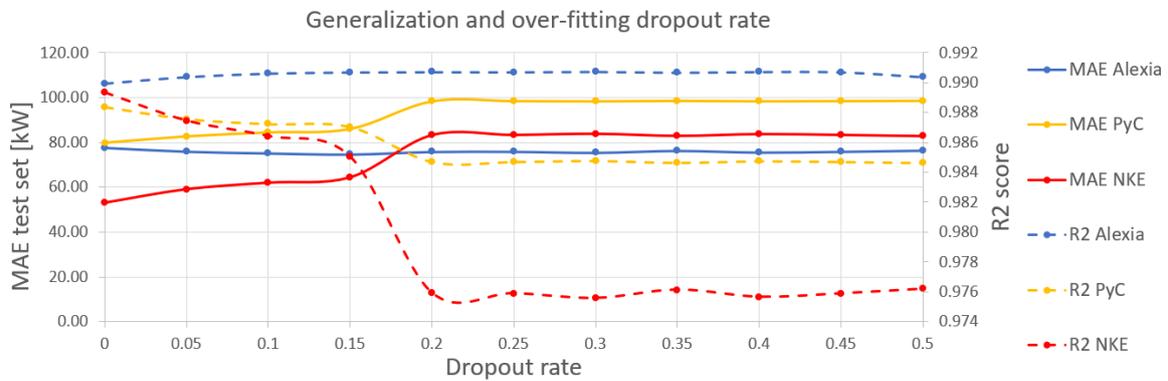
### Dropout rate

With the new optimized value of 40 for the early stopping epoch number, the dropout rate is optimized. As large dropout rates (over half of the epochs dropped out per iteration [57]) are likely to result in under-fitting, the tested range for the dropout rate ranges from 0.0 to 0.5, with an interval of 0.05. This yields in 11 cases to be evaluated for which the resulting plot of the MAE and  $R^2$ -score are shown in Figure 6-5

While a higher dropout rate seems to improve the  $R^2$ -score for Prinses Alexia slightly, an increase in dropout rate results in worse performance for all other evaluation parameters.



**Figure 6-4:** Early stopping epoch amount grid search test dataset results, evaluated per farm individually.



**Figure 6-5:** Dropout rate grid search test dataset results, evaluated per farm individually.

The network without any dropout seems to perform best and the  $R^2$ -scores for all wind farms are closest to each other. For these reasons the dropout rate was set at 0. This also means the network lay-out does not change for drop-out rate, as was elaborated upon in Section 3-2.

#### Regularization parameter/weight penalty

With both the early stopping epochs and dropout rate set, the remaining regularization parameter is the weight penalty. This method comes in three forms, as was shown in Section 3-2; L1, L2 and both L1 & L2 regularization. All three of these were evaluated and no significant difference was found between application of L1 or L2 regularization. Applying both, however, resulted in a consistent worse performance compared to L1 or L2 individually. As the resulting trend for L1 and L2 are similar, only the L1 regularization is shown (Figure 6-6).

For very small weight penalty, no significant difference is noted. However, from values of  $10^{-7}$  and higher, the performance of both PyC and NKE worsens, despite minor improvement for Prinses Alexia in both MAE and  $R^2$ -score. At a value of  $10^{-7}$  the evaluation parameters for each wind farm are also closest to each other. Therefore the regularization parameter was set at  $10^{-7}$ .

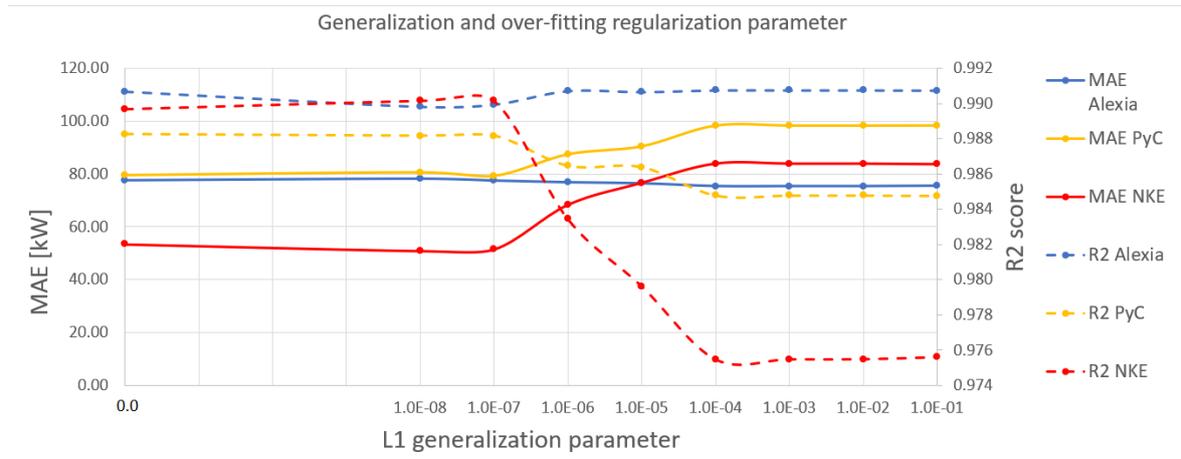


Figure 6-6: Weight penalty search test dataset results, evaluated per farm individually.

## 6-4 Discussion on optimized models

The above presented results were for the BCR model as this requires the most hyper-parameters to be determined. However, the same procedure is done for the other two models, except that the regularization (with exception of early stopping) is not considered. The goal of the other two models was to model the measured results closely, without obtaining an over-fitted model. The relevant grid searches are shown in Appendix F. The resulting model lay-outs are shown in Table 6-3. While the PCV model and PCA model are generally the same, the input and output to the 'sensitivity' model is standardized to a domain of  $[0, 1]$ . This removes any sensitivity from the size of connection weights, for example, as each input parameter has the same range.

**Table 6-3:** Grid search parameters for regularization parameters

Hyper-parameter	Business case refinement model	Performance correlation assessment model	Power curve verification model
Standard architecture (nodes/layer)	[14 - 8 - 12 - 1]	[9 - 14 - 8 - 1]	[9 - 14 - 8 - 1]
Corrected architecture (for dropout)	[14 - 8 - 12 - 1]	[9 - 14 - 8 - 1]	[9 - 14 - 8 - 1]
Internal unknowns (corrected unknowns)	353 (353)	341 (341)	341 (341)
Max. epochs	4000	4000	4000
Batch size	256	1024	1024
Test part of full dataset	25%	25%	25%
Layer activation functions	ELU	ELU	ELU
Output layer activation function	Tanh	Tanh	Tanh
Optimization parameter	Adam	Adam	Adam
Learning rate	0.0001	0.0004	0.0004
Loss function	MAE	MAE	MAE
Early stopping epochs	40	140	140
Drop-out rate	None	None	None
Regularization L1 penalty	$10^{-7}$	None	None

# Model results and applicability

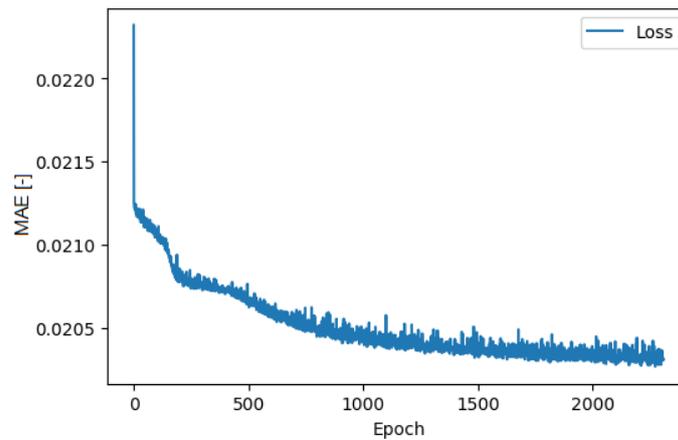
Optimizing the hyper-parameters and setting up the model was discussed in the previous chapter only for the Business Case Refinement (BCR) model, different optimal settings were found for the Power curve verification (PCV) and Performance Correlation Assessment (PCA) model. The goal and therefore results for each model are different for each other and are treated subsequently in this chapter. First the PCV model is discussed, followed by the PCA model, which is similar in settings as the PCV model since both focus in a single wind farm as input. Third is the BCR model, which is of secondary interest due to scarcity of combined (training) data from multiple wind farms. Finally some limitations on the methodology is discussed.

### 7-1 Power curve verification model

Since the PCV model is applied to single wind farms/turbines only, the model should be fitted to this farm/turbine and less generalization is required. It should not be omitted completely, as over-fitting remains a risk when using machine learning. For evaluation this means a single Mean absolute error (MAE) and  $R^2$ -score is available for each setting; for this single farm/turbine itself. The optimal hyper-parameter setting is therefore the optimal combination of these two characteristics. Furthermore, as the datasets for Pen y Cymoedd (PyC) and Norrekær Enge I (NKE) are less than 18 times the amount of internal unknowns, these farms are not yet evaluated in this model as this proved to be too small to consistently find the global optimum, the solution is still too dependent on the randomness in the solver. Therefore only Prinses Alexia is currently evaluated for the PCV model. For some qualitative comparison and evaluation of the model, also PyC is evaluated with the set network settings. However, these results should only be used as a rough comparison and not as intrinsic results.

## Training

The PCV model does not have generalization with the exception of early stopping to speed up the training process and reduce risk of over-fitting on the training data. With the setting as set in Table 6-3 and using the Prinses Alexia dataset as input, the resulting training error behaves as shown in Figure 7-1. Note the MAE, which was used as loss function, is based on the normalized output as used in the network and therefore dimensionless and not in kW. The amount of epochs required for training is about 2300 (changes for every training due to stochastic nature of solver), so the set amount of maximum epochs is not reached and the convergence criteria is met.



**Figure 7-1:** PCV model training error/loss function history.

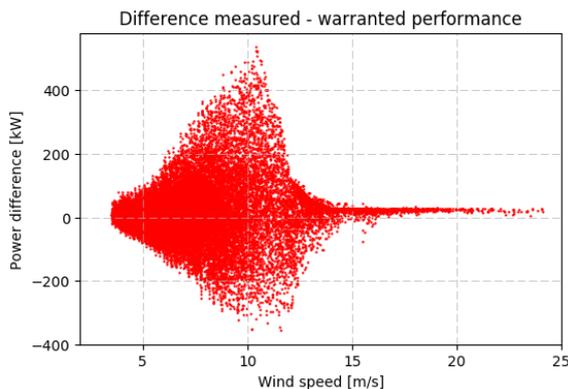
## Model validation and performance

With the model trained with 75% of the Prinses Alexia dataset, the remaining 25% is used as a test set to evaluate how the model performs with data it has not seen before. This validation is especially important as there are no (tangible) physics in the model, mainly a lot of statistically obtained relations. After validation, the performance of the model is evaluated and compared to the current method of the Warranted power curve (WPC).

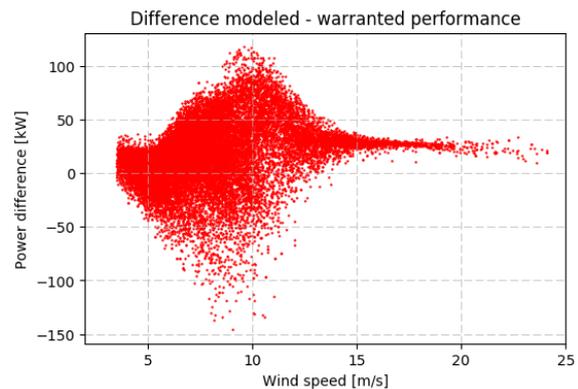
### Model validation

For validation the deviation or difference from the warranted power curve for both the measured as well as the modeled performance is plotted. This could also be done with both the modeled and warranted power curves, but this proved to be less suitable for comparison as both were 'just' a band following the warranted power curve and no distinct trends could be identified. Figure 7-2 and Figure 7-3 show this difference from the WPC for the measured ( $P_{measured} - P_{warranted}$ ) and modeled performance ( $P_{modeled} - P_{warranted}$ ) respectively. Important to note is these figures plot difference from *warranted performance*. Note the y-axis has a different range in both figures, meaning the model flattens the influence of the added parameters. This indicates the model is less likely to be over-fitted, which would be the case

if there was a (more or less) one-on-one correspondence. Having a smaller output range indicates a more regularized model, which is preferred as it is less over-fitted. When comparing both graphs, some characteristics can be observed. In both graphs the shift from the WPC increase for higher wind speed until about the knee of the power curve. Then the power difference range quickly decreases and the performance is slightly above warranted power when pitching of the blades sets in. It can also be observed that the model does not seem to be fitted to the more sparse regions in the measurements, such as the top and lower right in Figure 7-2. These regions have less data samples and therefore have less weight/importance during training of the model.



**Figure 7-2:** Performance deviation of measured power from warranted power for Prinses Alexia test subset.

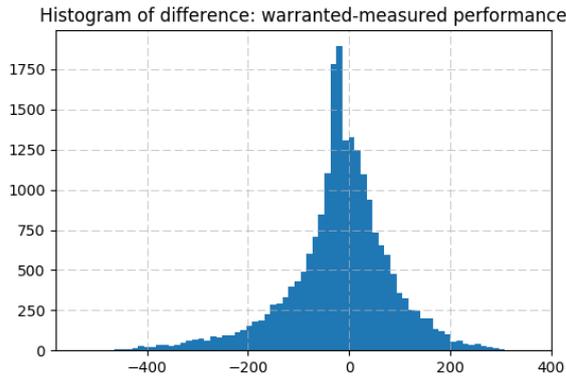


**Figure 7-3:** PCV model performance deviation of modeled power from warranted power for Prinses Alexia test subset.

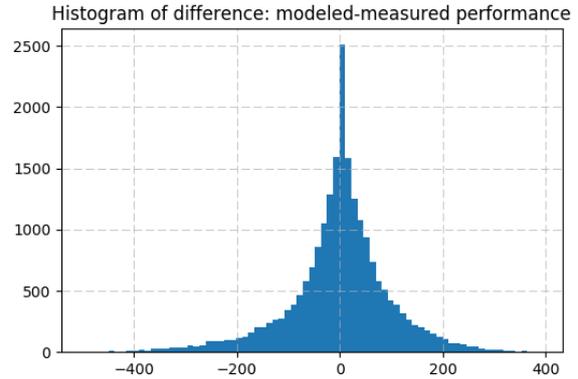
### Model performance

For validation of the model, as just discussed, the warranted power curve was used as baseline and the difference from the WPC was plotted. To evaluate the performance of the PCV model and compare it to the performance of the current method of using the WPC, both methods are compared to how the turbines actually performed; to the measured performance. It should be stressed Figures 7-2 and 7-3 are therefore very different from Figures 7-4 and 7-5! Although the difference between Figure 7-2 and Figure 7-4 is only a change of minus sign. Method performance comparison (WPC and the model) is first done by plotting the histogram of the difference from this measured performance. This difference is therefore:  $P_{predicted} - P_{measured}$  for all samples in the test case. The histograms for the WPC method and neural network model are respectively shown in Figure 7-4 and Figure 7-5. As can be seen when comparing both histograms, the model (right) is an improvement with respect to the WPC method (left). The spread is lower as there are less outliers with high absolute values of the difference from measured power. Furthermore the modeled peak is centered at approximately 0 and is significantly higher. Both indicate a higher accuracy and precision of the model with respect to the WPC method.

Besides visual comparison, the difference from measured power for both methods can be quantified through some statistical parameters. A selection is made of the MAE,  $R^2$ -score and Root mean squared error (RMSE), shown in respectively Equations 6-1, 6-2 and 6-5. The



**Figure 7-4:** Histogram of predicted power using WPC minus measured power for Prinses Alexia test set. Plotted with 75 bins.



**Figure 7-5:** Histogram of predicted power using modeled power minus measured power for Prinses Alexia test set. Plotted with 75 bins. Modeld with PCV model.

resulting values for both methods are shown in Table 7-1. While only the dataset for Prinses Alexia is considered large enough to consistently train a sufficiently accurate model, the model has also been trained and tested on the PyC dataset. This dataset is too small, however, to obtain valid results from and therefore should only be considered for this comparison as different training runs might result in very different models as the optimizing function is likely to find a different local minimum in the solution domain. But even though the amount of samples is considered too low, still a local minima is found and the resulting model is some indication of how well such a model will perform (even though a better minimum might exist). The results for PyC are shown on the right of Table 7-1.

**Table 7-1:** Performance of PCV model results vs. WPC for Prinses Alexia and PyC test dataset.

Parameter	Prinses Alexia			Pen y Cymoedd		
	Warranted PC	Modeled PC	Change w.r.t. WPC	Warranted PC	Modeled PC	Change w.r.t. WPC
R <sup>2</sup> -score	0.9907	0.9916	+0.1%	0.9863	0.9901	+0.4%
RMSE	105.46 kW	99.96 kW	-5.2%	125.43 kW	106.54 kW	-15.1%
MAE	74.51 kW	69.14 kW	-7.2%	96.02 kW	73.03 kW	-23.9%

As can also be seen in this table, the model outperforms the WPC method for both Prinses Alexia and PyC. The R<sup>2</sup>-score improves slightly, but especially the errors RMSE and MAE are 5.2% and 7.2% lower respectively. Together with the histogram in Figure 7-4 this shows the model is an improvement in predicting turbine performance for certain atmospheric conditions. Also for PyC the model is an improvement on the WPC, as RMSE and MAE improve by 15.1% and 23.9% respectively. The results for PyC should, however, be taken lightly as mentioned earlier.

## Practical use

With a functioning model for a single wind farm/turbine, the question arises what is the academic or practical use of it? While long-term energy yield predictions are not relevant

anymore (as there is no need for a business case energy yield prediction for existing wind farms), turbines are constantly monitored and performance evaluated to identify outliers from the power curve. This model or method could be the base of a monitoring tool able to instantaneously and continuously identify and flag outliers. From the training data and measured atmospheric conditions (if a met mast or Light Detection And Ranging (LiDAR) is present) expected performance for relevant turbines can be modeled. The (instantaneous) measured performance of the turbine can then be compared to this modeled performance and when a certain set deviation benchmark, or error, is exceeded that bin/moment is flagged. This indicates further investigation of the cause of the error is advised.

## 7-2 Performance Correlation Assessment model

The Performance Correlation Assessment (PCA) model, aimed to investigate individual influence on turbine performance, is very similar to the PCV model as both focus on a single wind farm and have similar goals. Both intend to accurately model turbine power curve for multiple parameters, without generalization, on this single turbine. Therefore the model settings are the same as the previous model. The only difference from the PCV model is that all input and training output is standardized between 0 and 1. The training target dataset (power deviation from warranted power) is therefore first normalized, as mentioned in Section 5-5, after which it is standardized between 0 and 1. Having all input and output range between 0 and 1 allows for Numerical sensitivity analysis (NSA) and therefore the individual variable sensitivity/correlation along the power curve, incorporating the variable's magnitude. Because only the input ranges vary, the validation results are exactly the same as the validation of the PCV model (Figures 7-2 and 7-3). It is therefore assumed the PCV model validation is sufficient validation of the model settings, while the model output itself is used for validation with literature and measurement, as far as possible.

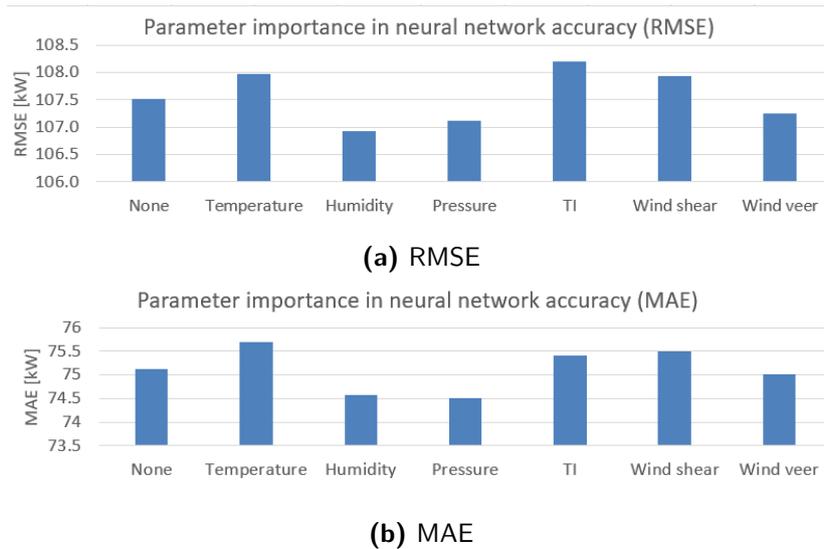
### Input parameter importance

The accuracy of the neural network allows for a relative quick analysis of relative parameter importance. The model trains a network to relate input parameters to the output parameter, resulting in a network with a certain accuracy, quantified through MAE or RMSE (for example). If the same procedure would be done with one input parameter less, this would result in a model with a different accuracy. Therefore by assessing changes in accuracy for training with a certain input parameter left out of the network, the relative importance of the parameter can be assessed. To obtain an accurate ranking this procedure is done for all input parameters in 15-fold, except the wind speed, where the average RMSE and MAE of these trained networks is shown in Table 7-2 and visualized in Figure 7-6. The procedure and purpose for n-fold training has been discussed earlier in Section 6-3. As training takes considerably less computational time and fewer training runs are required, the n-fold simulation amount is set three times higher than used in the previous chapter, resulting in 15-fold. A higher RMSE, MAE indicates the accuracy of the model is worse without this parameter, therefore the parameter can be considered relatively important. In Table 7-2 the average ranking of the RMSE and MAE ranking is shown in the right column.

**Table 7-2:** PCA model accuracy after leaving out single input parameter, including ranking of parameter importance.

Left out parameter	RMSE [kW]	MAE [kW]	Average ranking from RMSE and MAE	Importance ranking
None	107.53	75.13	-	-
Temperature	107.97	75.69	1.5	1
Humidity	106.93	74.57	5.5	5
Pressure	107.13	74.51	5.5	5
Turbulence intensity (TI)	108.19	75.42	2	2
Wind shear component	107.93	75.49	2.5	3
Wind veer	107.25	75.01	4	4

Interestingly it seems incorporating the wind veer results in a slight decrease in performance, as the RMSE and MAE is slightly lower when this is left out of the input parameters. The same holds for the pressure and humidity, which could possibly be replaced with density. However, as the change is only minor and this is only tested on just a single wind farm it would be too early to conclude leaving out these parameters would result in an overall improvement of this model. It should be kept in mind, however.

**Figure 7-6:** Change of PCA model accuracy with one parameter removed from the input parameters. Accuracy shown in (a) RMSE and (b) MAE

From this analysis the temperature turns out to be the most important parameter -besides wind speed- on turbine performance and can be considered to act primarily through the density. Slightly less important is the turbulence intensity, followed by the wind shear component. Wind veer comes in fourth place, followed by the humidity and pressure. The low relative importance of the pressure is slightly surprising as this also influences the density almost linearly. However, the minor changes in pressure for example (ranging about 950 hPa to 1050 hPa, so a range of  $\pm 5\%$  is likely the cause of the low importance. For comparison, temperature ranges between  $0^{\circ}\text{C}$  and  $30^{\circ}\text{C}$  (average about  $10^{\circ}\text{C}$ ) and therefore has a larger relative variation.

### Individual parameter numerical sensitivity

A useful aspect in assessing the influence of the influence of individual parameters on the turbine performance is the numerical sensitivity, comparable with the *Pearson* correlation coefficient. The numerical sensitivity indicates how a single parameter is correlated to another but, contrary to the Pearson correlation coefficient, does not assume the dataset(s) to be normally distributed. Therefore Numerical sensitivity analysis (NSA) is preferred to the correlation coefficient. To perform NSA, determining the sensitivity between input  $x_i$  and output  $y_k$ , the following steps have to be taken. [62] The expected value  $E$ , calculated in point 5, indicates the average effect of a change of input parameter  $x_i$  to output parameter  $y_k$  and represents the numerical sensitivity.

1. Arrange all test samples  $p$  in ascending order.
2. Split the full set in  $G$  groups  $g_r$  (more or less equally large).  $G = 30$  (meaning 30 groups) is deemed a sufficiently large for most cases. [62]
3. For each group determine the mean for the input parameter ( $\bar{x}_i$ ) and the mean of the output parameter ( $\bar{y}_k$ ).
4. Determine the NSA index per group using Eq. (7-1).

$$\text{NSA}_{ik}(g_r) \equiv \frac{\bar{y}_k(g_{r+1}) - \bar{y}_k(g_r)}{\bar{x}_i(g_{r+1}) - \bar{x}_i(g_r)} \quad (7-1)$$

5. Determine the sensitivity in terms of expected value, using Eq. (7-2).

$$E(\text{NSA}_{ik}(g_r)) = \sum_{r=1}^{G-1} \text{NSA}_{ik}(g_r) \frac{\bar{x}_i(g_{r+1}) - \bar{x}_i(g_r)}{\bar{x}_i(g_G) - \bar{x}_i(g_1)} \quad (7-2)$$

with  $\bar{x}_i(g_G)$  is the mean value of variable  $x_i$  in the last group  $g_G$ .

However, the correlation/sensitivity of a variable might be different along the power curve; for varying wind speed. Furthermore, the magnitude of the investigated parameter might also result in a different sensitivity on the turbine performance. To combine both these (possible) influences in the parameter sensitivity investigation NSA is performed for datasets which are binned for both wind speed, as well as a range of the investigated parameter. As this requires an extensive database, far larger than the available dataset of Prinses Alexia, a generic dataset is generated to be used as input of this model. In Appendix G both numerical sensitivity as well as the Pearson correlation coefficient are performed for the dataset of Prinses Alexia, for both comparison of the method as well as to show the importance of the dataset size. For a sufficiently large dataset of input parameters where only the wind speed and one other variable are varied, the turbine performance is modeled. All other parameters are kept constant. With this extensive database trained from Prinses Alexia, the sensitivity of a single parameter, binned for several ranges of magnitude, is calculated along the power curve. Furthermore for each binned range of the investigated parameter a separate power curve can be constructed and these multiple power curves can be plotted to see the influence of the magnitude of the investigated parameter.

The generic dataset to be constructed consists out of 100.000 samples. The wind speed and one single parameter are varied, while the other parameters are set as constant at about the average value of the training set. The range of the parameter to be investigated is slightly smaller than its range in the training dataset, as neural network tends to have problems for data beyond the training data since this is not trained for. Also the amount of training samples near the range borders is typically scarce. These constant values and ranges are shown in Table 7-3. The full range of the investigated parameter is split into 4 roughly evenly sized bins and the wind speed bin size is set at 1 m/s. As mentioned, for each of these bins the numerical sensitivity is calculated and plotted along the warranted power curve for easy comparison with each stage on the power curve.

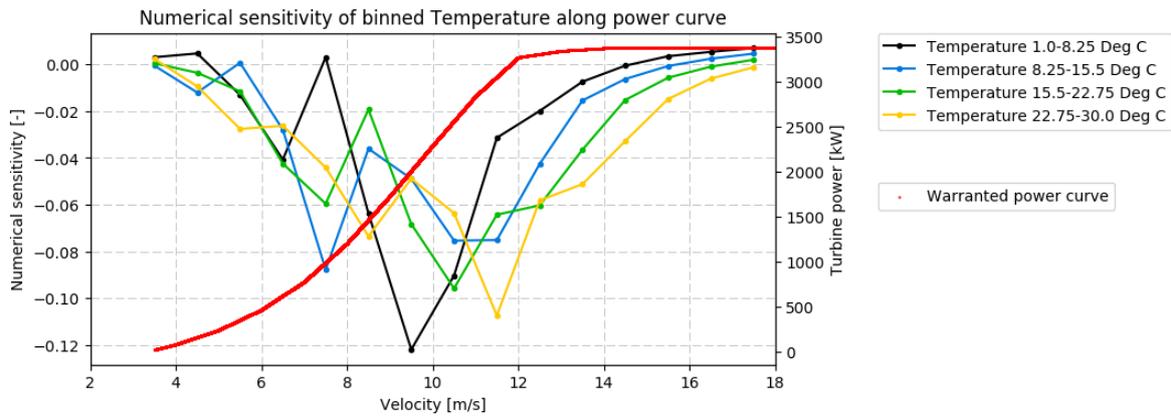
**Table 7-3:** Constant value and range of evaluated parameters, trained using Prinses Alexia dataset.

Parameter	Constant value	Range
Wind speed	-	3.5 m/s - 18 m/s
Temperature	10°C	1°C - 30°C
Humidity	80%	35% - 100%
Pressure	1000 hPa	970 hPa - 1030 hPa
Turbulence intensity	8.5%	2% - 25%
Wind shear component	0.20	-0.15 - 1.5
Wind veer	0.13°/m	0°/m - 1.1°/m
Generated dataset size: 100.000		

It would be preferred to also do a similar analysis to data which was actually measured, not modeled, but as mentioned earlier, this would require a vast amount of data, which is not present at the moment of this writing. But as is also mentioned in Appendix G, ranges for other parameters than the parameter to be investigated have to be  $\pm\sigma$  from the mean, otherwise the dataset would be too small to even split into 2 magnitude ranges of the parameter to investigate. Still the binned power curves for single parameters are plotted for 3 magnitude ranges and are compared to the binned power curves from the modeled results. These are shown in Appendix H and can be used as crude validation by comparing global trends. Discussed next are the binned numerical sensitivities of each parameter individually, in order of importance as shown in Table 7-2.

### Temperature

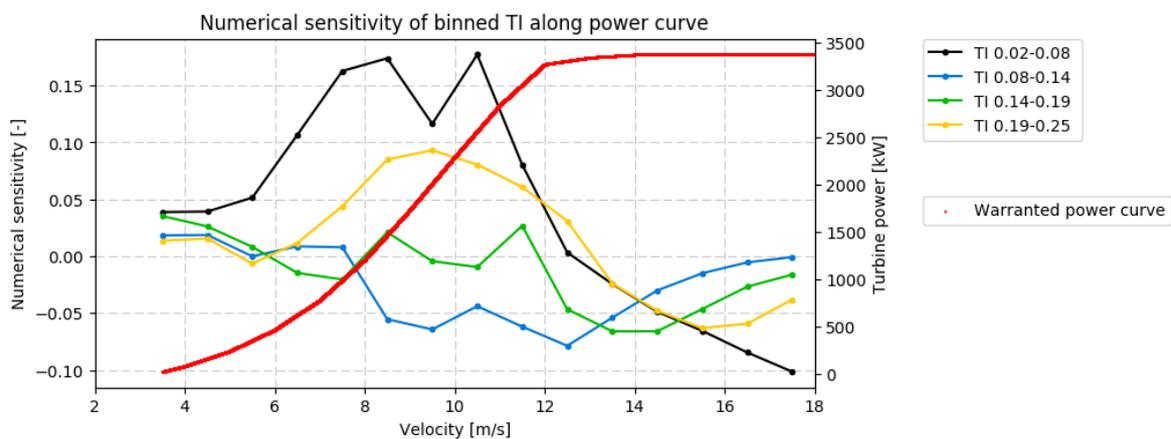
The numerical sensitivity for the temperature for several ranges and along the power curve is shown in Figure 7-7. The first trend which can be noticed is the sensitivity for the difference binned power curves are close to each other, indicating the magnitude of the temperature does not influence the sensitivity, or correlation, much. The second noticeable trend is the clear negative correlation of the temperature. Higher temperature therefore results in relative lower performance, with the effect mitigating when the controller sets in, near the constant power part of the power curve. The influence of the temperature can be directly found in the density, where temperature has a negative impact on density, while density has a positive influence on turbine performance.



**Figure 7-7:** Numerical sensitivity along power curve, singled out for ambient temperature. The neural model is trained for Prinses Alexia.

Turbulence intensity

The numerical sensitivity along the power curve using a generated dataset, varying only for TI, is shown in Figure 7-8 where several different impacts are noticed. For lower wind speeds, the TI has a small positive impact on the performance of the turbine, no matter the magnitude of the turbulence intensity. For higher wind speeds, the magnitude does seem to matter, however. Low TI tends to have a stronger, positive effect on performance, while high TI (above 0.19) also positively influences the performance. For medium turbulence intensity, however, the effect on turbine performance is both weaker and negative. For high wind speed, from about rated wind speed, the effect of all magnitudes of the TI has a -more or less similar for all- negative impact on the turbine performance. Both the positive impact for low wind speed and negative impact for high wind speed are in line with literature.

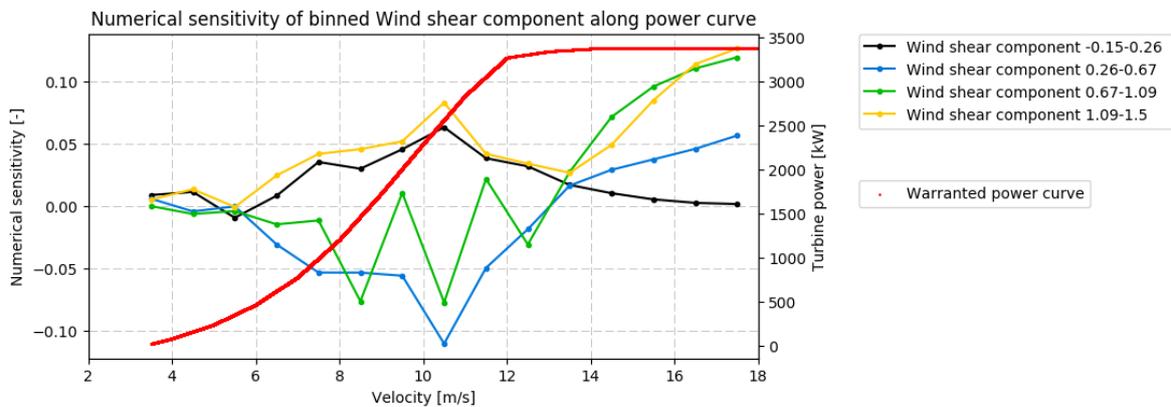


**Figure 7-8:** Numerical sensitivity along power curve, singled out for TI. The neural model is trained for Prinses Alexia.

## Wind shear

Numerical sensitivity for the wind shear along the power curve is shown in Figure 7-9. Here two main trends can be seen. The first is a similar trend between very low and very high wind shear components. Both (black and yellow) lines have a small positive relation with turbine performance. The two middle (blue and green) bins are also quite similar, but have a slight negative correlation. This indicates a positive parabolic influence of wind shear on the impact on turbine performance

Secondly, for (very) high wind speed, the impact tends to become more linear, where high wind shear has the largest positive correlation, with lower wind shear components having lower correlation. Especially the first trend is in line with the found impact by Walter et al., where wind shear has a parabolic influence, with high and low wind shear having a positive impact on performance, while in between there is a negative impact on turbine performance, as is shown in Figure 2-6. [7]



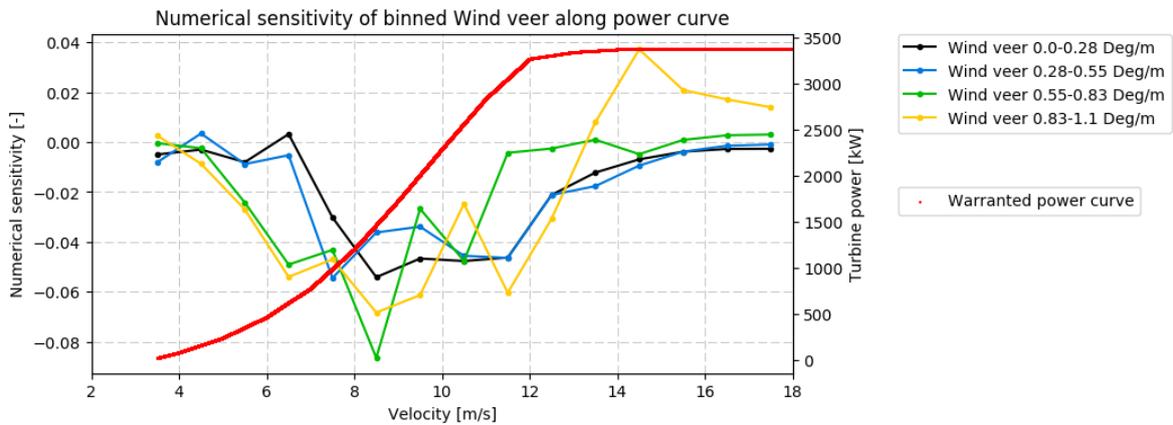
**Figure 7-9:** Numerical sensitivity along power curve, singled out for wind shear component. The neural model is trained for Prinses Alexia.

## Wind veer

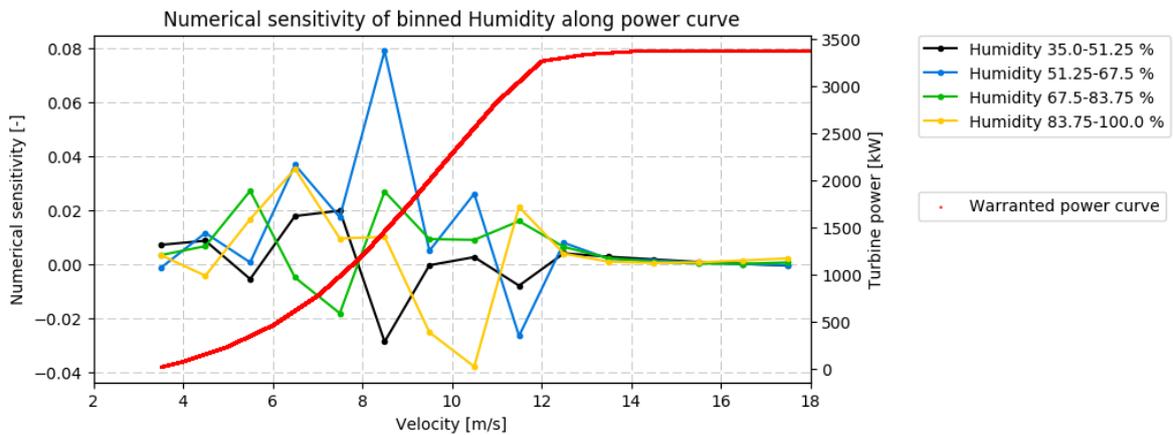
The impact of wind veer along the power curve, for different magnitudes, is shown in Figure 7-10. As all binned magnitudes are close to each other, the magnitude of the wind veer does not seem to be much of an influence on the correlation between wind veer and turbine performance. As expected the influence of wind veer is slightly negative, resulting from the deviation from the ideal inflow angle over the rotor.

## Humidity

The sensitivity of the humidity on turbine performance is shown in Figure 7-11, but no clear trends are visible. The magnitude of the humidity does not seem to matter much, as for all 4 binned curves, the sensitivity has no clear impact in the turbine performance. All binned curves switch multiple times between small positive and small negative influence, which could indicate the humidity is almost uncorrelated to the performance.



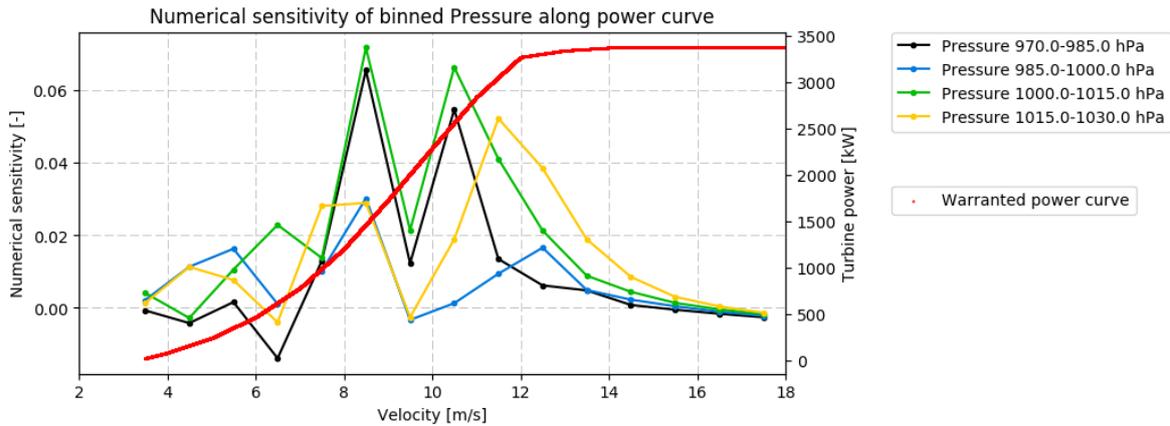
**Figure 7-10:** Numerical sensitivity along power curve, singled out for wind veer. The neural model is trained for Prinses Alexia.



**Figure 7-11:** Numerical sensitivity along power curve, singled out for humidity. The neural model is trained for Prinses Alexia.

## Pressure

Finally, the sensitivity of the ambient pressure to the turbine performance is shown in Figure 7-12. While no clear distinction in sensitivity between the different magnitudes of the pressure are visible, the overall trend is that the pressure has a slight positive effect on the turbine performance. This can be explained through, again, the density on which the pressure has a positive impact.



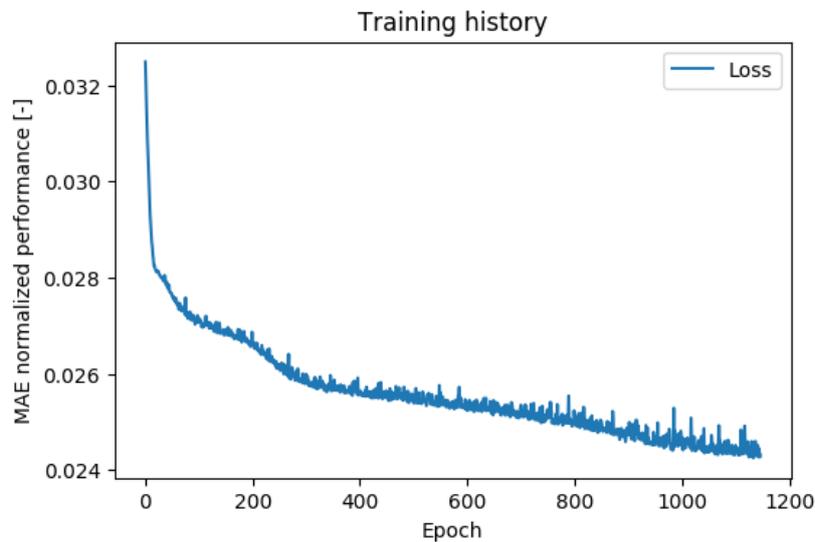
**Figure 7-12:** Numerical sensitivity along power curve, singled out for ambient pressure. The neural model is trained for Prinses Alexia.

### 7-3 Business Case Refinement model

The goal of the Business Case Refinement (BCR) model is to refine the long-term performance prediction of future wind turbines to external atmospheric variables. To remove dominance of a certain wind farm in the training dataset, the contribution of each wind farm to this dataset is equal. The downside of this is that it restricts the size of the total dataset to the size of the smallest dataset, NKE in this case. At the writing of this thesis this results in about 38 times more training samples than unknowns, which is considered to be sufficient to rule out dependency from stochastic behavior of the optimizing/training function. Despite it can be assumed the total amount of training samples is sufficient, there is a lack of diversity of wind farms as only 3 farms are used, among which 2 have -more or less- the same site conditions. While this limits the accuracy and general application of the results at the moment of this writing, the 'intermediate' results compared to the warranted power curve method, can be interpreted as a potential of the method. This should be kept in mind when evaluating the results and comparing these to the 'traditional' method of using the Warranted power curve (WPC).

#### Training

Training of this model was done with equal contribution of each wind farms Prinses Alexia, PyC and NKE, resulting in a total training set of about 13,530 samples for 353 internal unknowns. The error history during the training process is shown in Figure 7-13. As can be seen the required number of epochs is far smaller than the set maximum of 4000 epochs and it can be concluded the solution has converged.



**Figure 7-13:** BCR model training error/loss function history.

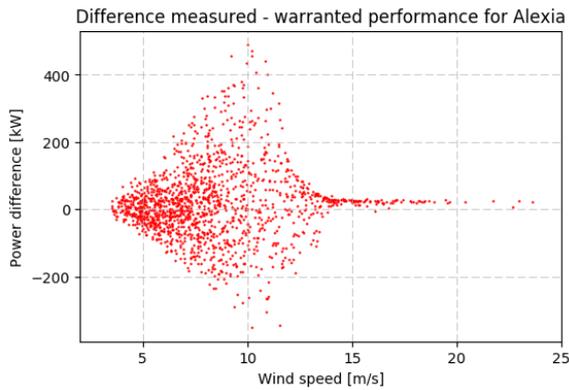
## Model validation and performance

In the same manner as with the PCV model, the trained model should first be validated whether the results actually match the measurements (to a certain degree), before evaluating the performance of the mode.

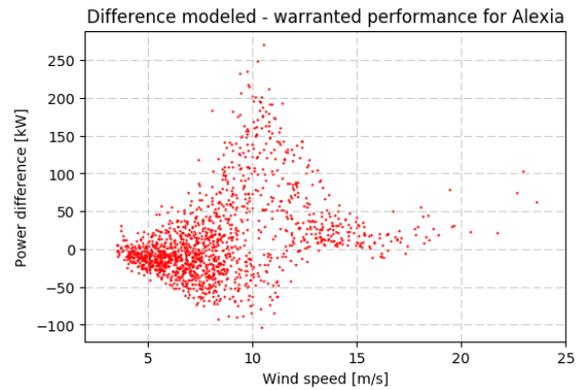
### Model validation

Again instead of plotting the resulting power curve from the model and measured power, the difference from the warranted power curve is plotted as this shows more trends and patterns, while plotting 'just' the power curve only shows a sparse band along the WPC. Only the test dataset is used for the validation analysis as the network has not been trained for this data, therefore providing for the most reliable comparison. The difference of both measured and modeled power from the warranted power curve ( $P_{measured} - P_{warranted}$  and  $P_{modeled} - P_{warranted}$ ) is shown for Prinses Alexia, PyC and NKE in Figures 7-14 to 7-19, with the difference of  $P_{measured} - P_{warranted}$  on the left and  $P_{modeled} - P_{warranted}$  on the right.

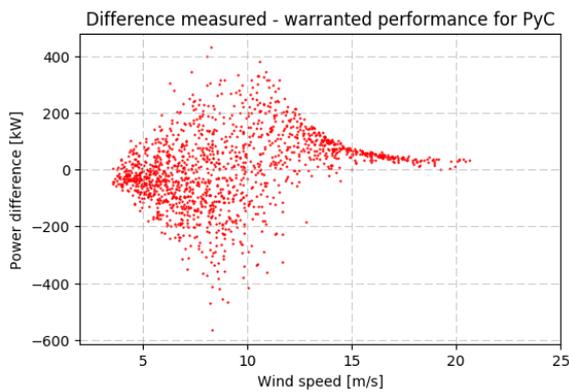
Looking at general trends and ranges, the model does incorporate some trends in the measurements. Comparison is somewhat difficult as the trends should be a combination of all three wind farms used for training. However, while the model seems to predict the general trends up to about 15 m/s rather well, for higher wind speeds there seems to be some incorrect modeling. The reason for this is, as is also discussed earlier, the lack of data especially in this wind speed range. While the total training set is assumed to be sufficient for consistent training, the amount of samples for high wind speed is significantly smaller than lower wind speeds. Therefore the trained network is likely to perform relatively worse for high wind velocities.



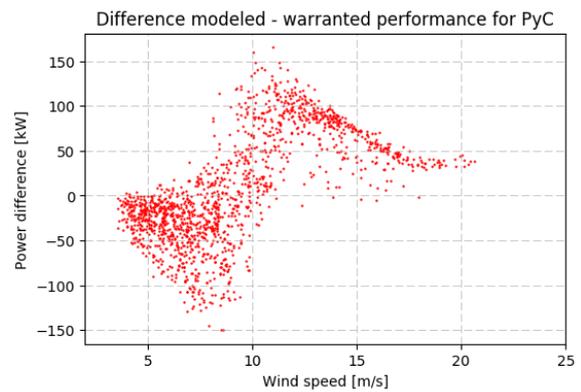
**Figure 7-14:** Difference between measured and warranted power for Prinses Alexia vs. wind speed.



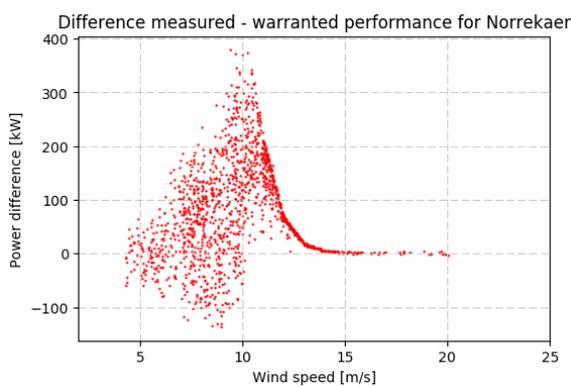
**Figure 7-15:** Difference between modeled and warranted power for Prinses Alexia vs. wind speed, modeled with BCR model.



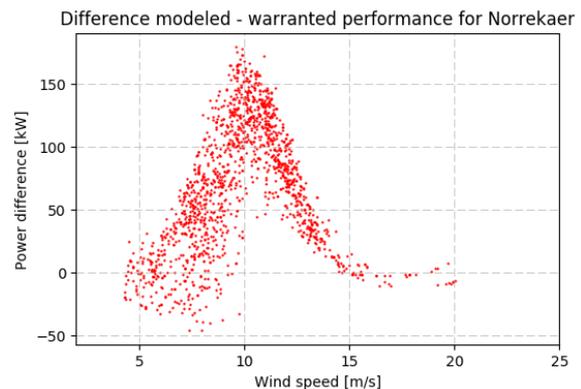
**Figure 7-16:** Difference between measured and warranted power for PyC vs. wind speed.



**Figure 7-17:** Difference between modeled and warranted power for PyC vs. wind speed, modeled with BCR model.



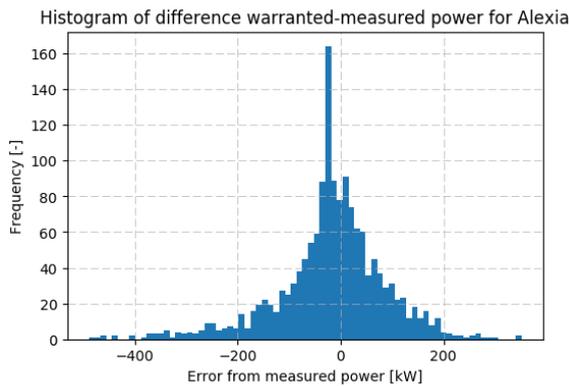
**Figure 7-18:** Difference between measured and warranted power for NKE vs. wind speed.



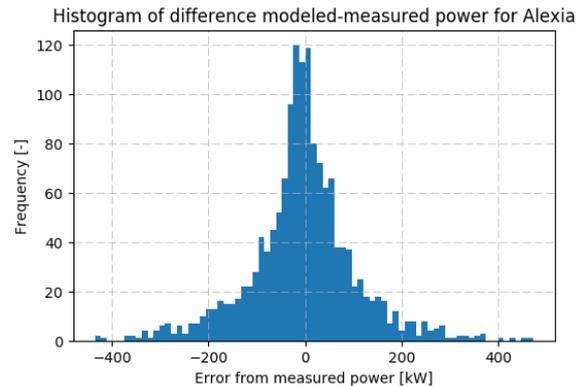
**Figure 7-19:** Difference between modeled and warranted power for NKE vs. wind speed, modeled with BCR model.

### Model performance

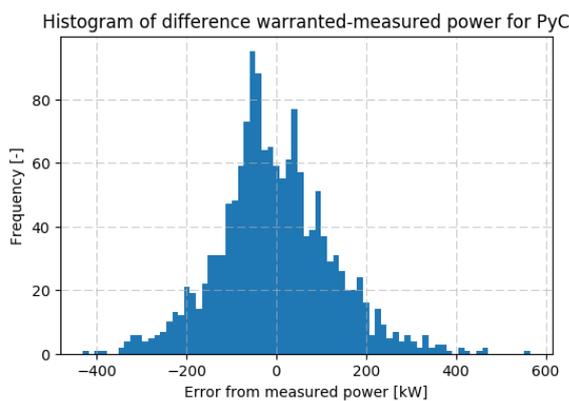
To evaluate the performance of the model for the three wind farms, again the histograms of the difference from the measurements for both methods (WPC and neural network model) are plotted. The histograms of the error between warranted power and measured power and the error between modeled power and measured power are shown in Figures 7-20 to 7-25, with the error between warranted and measured power on the left.



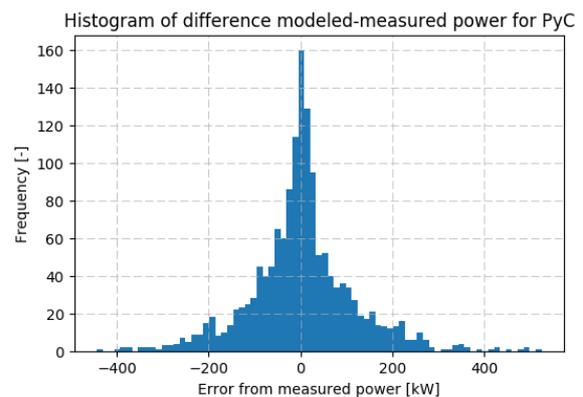
**Figure 7-20:** Histogram of error between warranted power and measured power for Prinses Alexia test set. Plotted with 75 bins.



**Figure 7-21:** Histogram of error between modeled power and measured power for Prinses Alexia test set. Plotted with 75 bins, modeled with BCR model.

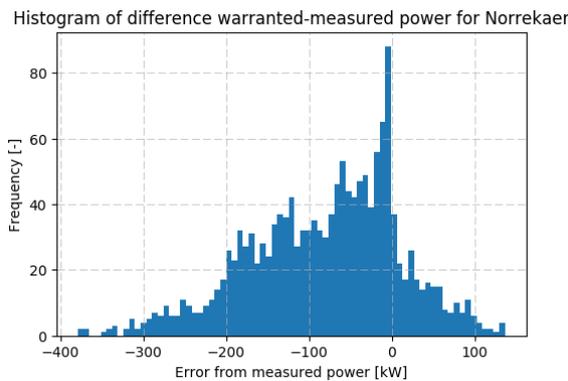


**Figure 7-22:** Histogram of error between warranted power and measured power for PyC test set. Plotted with 75 bins.

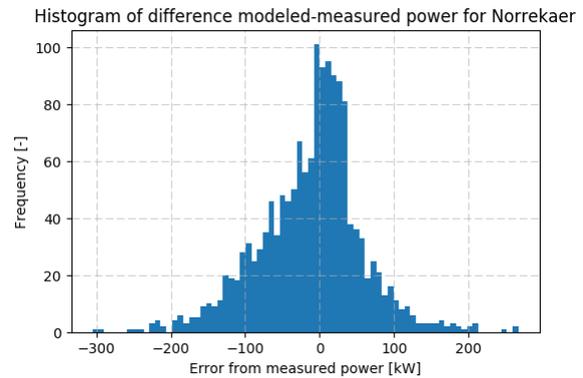


**Figure 7-23:** Histogram of error between modeled power and measured power for PyC test set. Plotted with 75 bins, modeled with BCR model.

When comparing the histogram for Prinses Alexia the error of the model seems a little more spread out, toward a more symmetrical distribution, also visible in the low error bins. These are lower than the highest peak from the WPC method, but spread over some bins with low error. For PyC a clear improvement can be seen. The peak is located in in the (near) 0-error bin and is spread out more symmetrically. The bins for errors close to 0 are also higher than the same bins in the WPC method, indicating more samples with a low error.



**Figure 7-24:** Histogram of error between warranted power and measured power for NKE test set. Plotted with 75 bins.



**Figure 7-25:** Histogram of error between modeled power and measured power for NKE test set. Plotted with 75 bins, modeled with BCR model.

For NKE several impacts can be seen. First of all, the warranted power curve under-predicts the turbine performance for the major part, with the largest peak occurring close to 0 error. The model again shifts the spread of the error to a more symmetric distribution. Remarkable is the large collection of high bins in the positive side of 0 error, which is obviously not due to the influence of the NKE training data. This is clearly influenced by the other two wind farms, where PyC is very likely to be the main cause of this trend. On a visual basis the model seems to be an improvement on the WPC method as there are higher bins for lower errors.

Since visually comparing the histograms is a limited method of comparing method performance, the performance of both is also quantified in again the  $R^2$ -score, RMSE and MAE. However, as this model should, in the end, yield in a single number of a long-term energy yield estimation, two other variables are also considered; the mean error and the percentile error of the total sum of energy predicted. While the MAE and RMSE determine the accuracy of the prediction on the actual measured power, the mean error and percentile error of the total sum of energy take into account some leveling out of positive and negative predictions. This is expected to happen (and does happen) when using the WPC, as sometimes the performance is over-predicted and sometimes under-predicted for the same wind speed bin. The goal of this model is, however, to be less dependent on this seemingly random spread within wind speed bins. In power curve verification the turbine performance is also merged over a whole time-frame, where again positive and negative performance deviations cancel each other out. These statistical properties are shown in Table 7-4 for Prinses Alexia, PyC and NKE.

## Practical use

The current state of the model is of limited use, due to the somewhat unilateral availability of wind farms; only 3 farms from which 1 has complex terrain. Despite this it does show a potential to refine the long-term energy yield estimations. The refinement model shows improvement for both Prinses Alexia and NKE compared to 'just' the WPC. Therefore some practical use of such a model is considered for the future. So how could such a model be used in practice then?

**Table 7-4:** Relative performance between WPC method and BCR model quantified for Prinses Alexia, PyC and NKE test sets.

	Prinses Alexia		Pen y Cymoedd		Norrekær Enge I	
	Warranted	Modeled	Warranted	Modeled	Warranted	Modeled
R <sup>2</sup> -score	0.9905	0.9899	0.9852	0.9882	0.9702	0.9900
RMSE	107.26 kW	110.64 kW	127.90 kW	114.10 kW	121.3 kW	70.46 kW
MAE	75.64 kW	77.64 kW	98.45 kW	79.04 kW	94.04 kW	51.85 kW
Mean error	-18.20 kW	-5.12 kW	-2.85 kW	3.48 kW	-80.68 kW	-11.74 kW
Percentile error sum total energy yield	-1.24%	-0.35%	-0.18%	0.22%	-5.08%	-0.74%

There are some requirements which are brought along in using a neural network. Besides -less important- practical requirement such as software, the model method requirements can be listed as:

1. Sufficient data is important and there is assumed to be sufficient data to train this model. However, more versatile data would increase the generalization of this model. Having data from a higher number of wind farms also decreases possible dominance of individual farms as influence from each wind farm is mitigated.
2. The input for the considered, new, wind farm should match the network input. When one or more input parameters are not available, this (or similar) model can still be used, but has to be retrained taking only the available input parameter into account. This also requires new hyper-parameter optimizations, however.
3. The range of the training data is to be the range of the variables. As the network is trained for a certain range for each variable, input beyond these ranges might result in unexpected output of the network. Neural networks are slightly infamous for their inability to model for input beyond the training input ranges.

These requirements considered, such a BCR model finds it applicability right after the performance prediction (e.g. annual energy prediction), but before corrections such as for farm lay-out, wake and icing. Individual warranted turbine performance is refined to the site conditions with individual turbine performance as output.

As the third requirement, limiting the input range to the training input range, is dependent on the farms used in training, it would be an improvement for the refinement estimation to make a division in, for example, terrain complexity or wind/turbulence classes as defined by the International Electrotechnical Commission (IEC). TI, for example, is influenced by the terrain complexity and more complex terrain has wider ranges of TI. Training individual models for each turbulence class would refine the model on these classes improving accuracy. However, the first requirement then limits the reliability and complexity of such models, as sufficient data should be available to train the networks and this has shown a limitation at the moment of writing of this report.

## 7-4 Limitations and shortcomings

Despite some promising results, especially for the PCV model, there are some limitations and shortcomings to the current method. Important limitations to consider are:

- The trained models can only be used for the parameters as provided as training data. Not only are these parameters all required, but also the measurement location and possible calculation method (e.g. for TI) should be the same. Differences will yield in incorrect output. The input parameter range is also limited to the training parameter ranges, as ranges beyond training input ranges might result in undesirable inaccuracy. This is a known limitation of neural networks.
- Input in the models should be pre-processed to a certain degree. Raw, unfiltered data can be used but is likely to yield in some inaccurate results. While a lot of rules can be automated, some human interaction should be required to assess this filtering and adjust where necessary. Also for these methods holds the *garbage in, garbage out* principle.
- The training of the network is in some degree stochastic dependent, mainly in the optimizing function. While this is no problem for large training sets, it is for smaller sets as has been described. Large training datasets are therefore a hard requirement. This also limits the model complexity in terms of network architecture.
- Since the available data was limited to three wind farms, there was limited possibility to validate the models from this thesis. Especially since the relevant dataset size of two of these wind farms are small at the moment of writing, it is only tested for a single wind farm. In the future more data is available, also when cheaper measurement methods like LiDAR are used more frequent and therefore more testing and validation will become possible.
- The current models have a refinement on the turbines warranted power as output, which is of course dependent on the warranted power curve itself. However, as this WPC is constructed by the manufacturer, slightly adjusted for certain site conditions and a trade-off between uncertainty and competitiveness. This bring along a certain degree of subjectivity, which makes this method slightly more uncertain and dependent on the 'opinion' of the manufacturer. This 'relative performance' should therefore be carefully monitored when using this method.
- While the PCV model is specified on a single turbine type, the BCR model takes data from multiple different turbines as input, which are also located in different terrain (complexity). Combining all available data, as was done in this project, results in a questionable model accuracy compared to using training data from similar sites only. If data availability allows it should prove to be beneficial to make divisions in aspects like terrain complexity, wind classes and turbine size/manufacturer/type. This requires a massive mount of data, however, and should currently be considered as infeasible.

# Conclusions and recommendations

## 8-1 Conclusions

Despite the somewhat limited availability of sufficiently large validation datasets from various wind farms, some conclusions about this machine learning approach on multivariate power curve modeling can be made. Furthermore for wind farm sites with certain similarity to Prinses Alexia the impact of certain external parameters on wind turbine performance is assessed using the same approach. Key conclusions can therefore be drawn for the three models: Power curve verification (PCV) model, a sensitivity analysis via this model, the Performance Correlation Assessment (PCA) model, and the Business Case Refinement (BCR) model for refinement of the long-term energy yield estimations for future wind farms.

### Power curve verification model

Using a neural network a model is constructed, validated and tested for Prinses Alexia wind farm and Pen y Cymoedd (PyC), where the dataset of PyC is considered to small for thorough validation and testing. The approach shows such a model is considerably more accurate than by only using the Warranted power curve (WPC) for estimation of the turbine performance. For Prinses Alexia this yields in Root mean squared error (RMSE) and Mean absolute error (MAE) improvements of 5% and 7% with respect to the warranted power curve method. These improvements were even larger for PyC, but due to the limited dataset size its validation is limited.

It can therefore be concluded this method is a valid approach in power curve verification *after* the verification period, performed by the third party, if the training dataset size allows. This would allow the wind farm operator to constantly monitor turbine performance with a future outlook for instantaneous flagging of outlying performances.

## Performance Correlation Assessment model

The model for investigating the individual impact of atmospheric influences on turbine performance is validated and can therefore be assumed to model the multivariate power curve, of the wind turbine for which it was trained, to a satisfying accuracy. This allowed for specific sensitivity analyses, singling out input parameters and thereby investigating their impact on turbine performance. Despite the site conditions of the investigated wind farm, Prinses Alexia, is not the most interesting terrain as it is flat with few obstacles, some new insights in the impact of certain atmospheric conditions can be derived. By providing the trained model with specific datasets, varying only one parameter which is to be investigated, some influences were found. These are, however, only valid for Prinses Alexia wind farm, testing on additional wind farm was not possible and the general applicability of the method can not yet be verified.

- The Turbulence intensity (TI) did have a negative, though small, impact on performance at low wind speeds. For higher wind speeds the positive effect on turbine performance was clearly noticed, as can be confirmed with literature.
- The effect of wind shear is considerably less than TI, but also non-linear as both high and low wind shear have a positive effect on wind turbine performance. Medium wind shear, however, has a slight negative impact on turbine performance.
- The ambient temperature did have a clear negative impact in turbine performance, which can be explained through the density in the wind power flux.
- Wind veer is also negatively correlated to the turbine generated power, which is caused due to the change in inflow angle over the rotor. Even through it is small, it is clearly noticeable.
- Relative humidity does not seem to have a distinct influence on turbine performance at Prinses Alexia wind farm.
- The ambient pressure has a slightly positive impact on the turbine power, which can also be explained through its slight positive influence on the air density.

## Business Case Refinement model

Even though the full training dataset size of wind farms Prinses Alexia, PyC and Norrekær Enge I (NKE) combined was deemed insufficient to accurately train and validate a general predictive model, current results were promising. Modeled turbine performance for PyC and NKE showed improved accuracy compared to the warranted power curve predictions. The modeled turbine performance for Prinses Alexia, however, was a slight decrease in terms of absolute errors, but the prediction was more symmetric, yielding in a more accurate prediction of the total energy yield over the evaluated time window compared to the warranted power curve prediction.

However, some remarks should be made regarding this model. As mentioned, dataset size was limited and therefore validation was not sufficiently possible. Furthermore, there was

significant difference in terrain complexity from wind farm to farm as PyC has very complex terrain, while Prinses Alexia and NKE are very flat. This leaves a recommendation about possibly splitting for terrain complexity as will be discussed in the next section.

## 8-2 Recommendations

As this approach shows potential, especially for the ever growing collection of data, there are some recommendations for future work and/or applications following this research. These are again split per different model application. One recommendation can be made independent on the type of model studied in this report, which is discussed first.

### General recommendation

In this report the base of each model was a neural network, which is itself an ideal black box. While having certain advantages, it clearly has some disadvantages. Further work could be looking into different machine learning applications, but also incorporating this non-physical model with a physical model. Certain aspects of wind turbine performance are researched extensively and can be modeled in a physical way, possibly using Adaptive neuro-fuzzy interference system (ANFIS) methodology. Then the black box approach can be attached to such a white box method, yielding in a sort of gray box model with some part physics, some part learning of existing data. Further addition to this is using (generic) optimization functions to also train or refine the physical model to empirical results.

Besides this combination, a simpler machine learning algorithm can be considered. Neural networks are typically considered as the more complex algorithms, but the constructed models do not actually approaches the deep part in deep learning. So a simpler model, for example Support vector regression (SVR), might also result in good results, but without the network complexity.

An application of this method could also be wake loss prediction. By comparing turbine performance in the free stream by performance of the same turbine in a range of wake conditions, an empirical relation might be found which might improve the wake losses. Current predictions still introduce quite some uncertainty to this loss and a machine learning method could yield in reduced uncertainty, for low computational cost.

### Power curve verification model

The PCV model has shown to be an improvement on estimating performance for a single wind turbine, based on its (verified) performance in its operational lifetime. However, in its current state it is of very limited applicability. Further development, especially with Supervisory Control And Data Acquisition (SCADA) systems could result in an application which could almost instantaneously flag (10-minute averaged) moments where a wind turbine does not perform as it should, based on the atmospheric conditions. A further requirement is the simultaneous availability of this atmospheric conditions, which is currently a limitation.

Besides an instantaneous error flagging application, this model/method can be a base of an in-company application of verifying the measured performance of the wind turbine. As typically

atmospheric data (via meteorological mast or LiDAR) is available in batches over a longer time window (typically months), easier verification of the turbine performance could become possible, to be performed after collection of the required datasets. This would give more insight in the condition of the wind turbine over its lifetime and might indicate prolonged issues with the turbine.

### **Performance Correlation Assessment model**

Validation of the PCA model can be done by training for/testing on more wind farms, where also the impact of individual parameters can be compared between farms. Possible coincidences in found dependencies or correlations might be found. Furthermore combining wind farm datasets might result in a general impact of individual parameters on turbine performance, which could be of interest especially for offshore sites, where terrain conditions are fairly similar.

Looking to other fields of application, similar methodology could in theory be used for load assessment, for example fatigue. By assessing the loads with certain conditions and turbine settings, this method might prove to be an improvement on current fatigue assessment models. This happens to be a field of interest, since at the Vattenfall R&D department similar statistical method, machine learning, applied to fatigue assessment is already being researched.

### **Business Case Refinement model**

Using the model for refining the long-term energy yield estimation did prove to be an improvement compared to using only the warranted power curve. However, due to a lack of sufficiently large datasets from additional wind farms its general use could only be limited tested. This limits both its validity, but also applicability as it makes less sense to use a model trained for flat terrain on highly complex terrain, for example. Should data allow, the generality of this model can be tested. Additionally it could prove useful to first split the datasets in either terrain complexity classes, wind classes or both and train and use/train separately for each class. More in depth studying of the influence of atmospheric parameters in turbine performance could reveal these correlations can also be classified in named classes. This will improve accuracy of each model, but would in the same time require a massive amount of data as each different class requires its own training database.

---

# Appendix A

---

## Wind farm Prinses Alexia

In this Appendix the site conditions of Prinses Alexia wind farm are shown in more detail, to provide a full picture of the wind farm. In Chapter 4 the site, site conditions and measurement instruments were briefly discussed and for the full picture some details mentioned there are repeated here.

### A-1 Wind farm site

Vattenfall wind farm Prinses Alexia is located in the South of the province Flevoland in The Netherlands as shown in Figure A-1. The terrain surrounding Prinses Alexia is flat and with very few obstacles; a few farms and scattered trees. Therefore no measurements prior to construction were performed on the specific location, but measurement of airport Lelystad (around 20 km North, with similar terrain) was used. Furthermore a SoDAR campaign was performed prior to construction of the wind farm. During power curve verification, the mean density and Turbulence intensity (TI) were measured at  $1.251 \text{ kg/m}^3$  and 8.9% respectively. [15]

Prinses Alexia consists of 36 wind turbines of type 3.4M104 manufactured by Senvion, some of the turbine characteristics are shown in Table A-1. The wind farm was opened in September 2013 and since then only one has been built, which was also used as reference mast -or permanent mast- for power curve verification. This means this mast is the only mast of Prinses Alexia used in this research. This met mast is paired to two wind turbines, named R01 and R02, which are closest to the met mast and conditions can be assumed equal (for a certain range in wind direction). The location of the met mast and the relevant turbines is shown Figure A-2, where the relevant turbines are colored in dark blue, in the center of the map, the met mast colored in white and all other are turbines colored in light blue. Note that the turbines in the orange circle are part of Prinses Alexia, but not all of the turbines of the wind farm. More turbines are located to the East, but are not relevant to this analysis. The relevant turbines R01 and R02 have been used for power curve verification by a third party and since this data is available, these turbines are used as input.

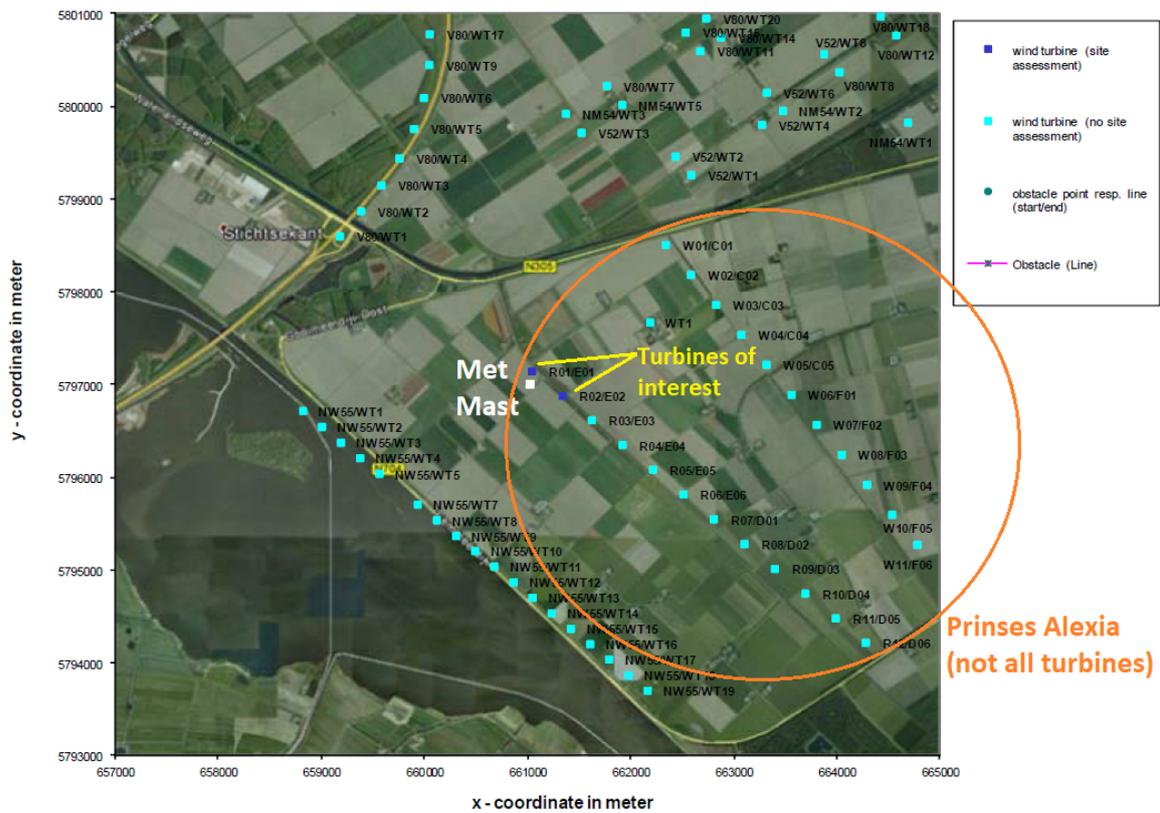


**Figure A-1:** Location of wind farm Prinses Alexia in The Netherlands. Map via Google Maps.

**Table A-1:** Turbine specifications Senvion 3.4M104, via [15]

Parameter	Value	Parameter	Value
Rated power	3370 kW	Rated wind speed	13.5 m/s
Hub height	98 m	Cut-in wind speed	3.5 m/s
Rotor diameter	98 m	Cut-out wind speed	25 m/s

Turbines in the Western and Northern neighborhood are of type V80 and V52 and have, according to the International Electrotechnical Commission (IEC) standard [14], no influence on both the turbines as well as the met mast, as the closest is at a distance of 2.4 km (direction 308°). In the South and West there are turbines of type NW55 and these are also considered not to have an influence, as the distance of the nearest turbine is 1.8km (direction 218°).



**Figure A-2:** Prinses Alexia location of met mast and relevant turbines. (from [15], edited by author)

## A-2 Measurement equipment

The met mast is located at a distance of 222m ( $2.13$  rotor diameter,  $D$ ) from R01 and 405m ( $3.89D$ ) from R02. Table A-2 shows the mounted measurement instruments on the met mast, as well as their altitude with respect to ground level.

**Table A-2:** Turbine specifications Senvion 3.4M104, via [15]

Measurement instrument	Altitude above ground level	Alignment with respect to North	Unit	Resolution/accuracy
Pressure sensor	95.8m	-	hPa	0.1 hPa/0.1%
Relative humidity sensor	96.3m	-	%	1%
Air temperature sensor	96.3m	-	°C	0.1°C/0.1%
Cup anemometer A4	46.4m	318°	m/s	0.1Hz
Cup anemometer A3	96.0m	308°	m/s	0.1Hz
Cup anemometer A2	98.0m	130°	m/s	0.1Hz
Cup anemometer A1	98.0m	308°	m/s	0.1Hz
Wind vane	46.3	134°	°	1°/0.1%
Wind vane	96.1	130°	°	1°/0.1%

### A-3 Power curve verification

Power curve verification (PCV) for Prinses Alexia has been performed by WIND-consult GmbH (WICO), who used their own Turbine power performance indicator (TPPI) to measure performance of the turbine. For an extended period of time the performance is assessed and on a regular interval the results are reported to the farm operator. For R01 and R02 power curve verification has been performed from 03/02/2015 up to 09/09/2017 by WICO and these results are provided to the farm operator, therefore available for use in this thesis. If this would not be the case, Supervisory Control And Data Acquisition (SCADA) data would be linked to met mast measurements, which will still be available if the met mast is bought by the farm operator.

### A-4 Measurement sectors

As discussed in Chapter 4 some wind direction bins are considered to have undisturbed upwind conditions (free stream), some in the wake of nearby turbines (wake stream) and the rest as non-similar or blocked. The corresponding wind direction sectors were provided and again shown in Table A-3. Figures A-3 and A-4 show these sectors for turbines R01 and R02 respectively.

**Table A-3:** Turbine wind sector ranges

	Free sector	Wake sector	Blocked sector
R01	163° - 348°	130° - 163°	348° - 68° (R01 - met mast )
	-	-	70° - 130° (R02 - met mast )
R02	164° - 281°	130° - 164°	281° - 343° (Non-similar conditions)
	343° - 348°	-	348° - 68° (R01 - met mast )
	-	-	70° - 130° (R02 - met mast )

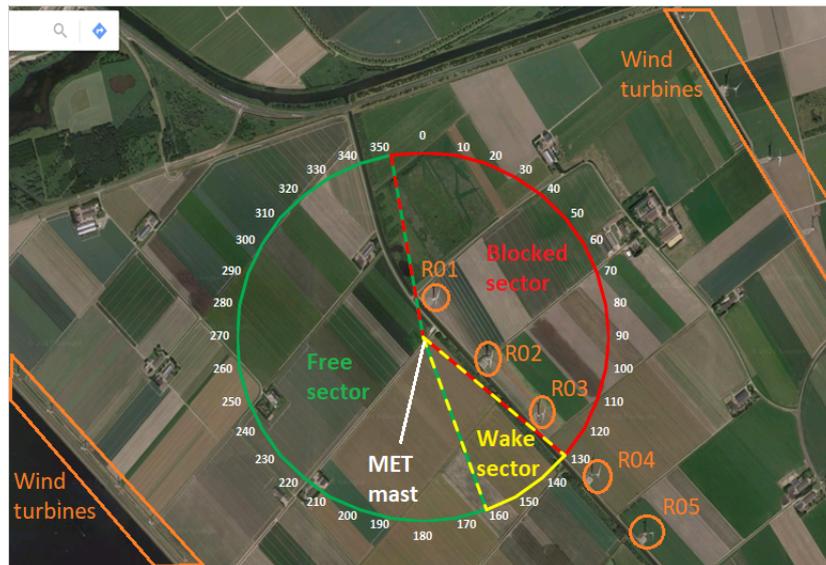


Figure A-3: Wind sectors for turbine R01 and paired meteorological mast.<sup>1</sup>

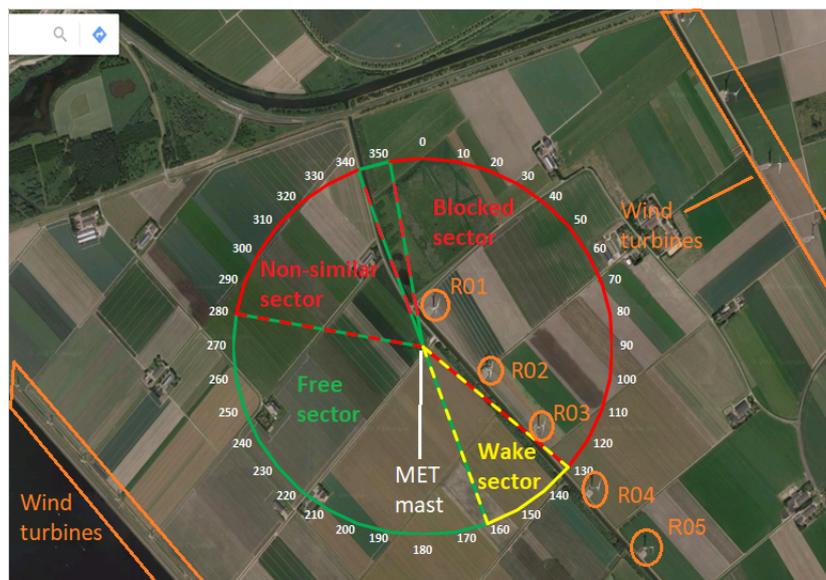


Figure A-4: Wind sectors for turbine R02 and paired meteorological mast.<sup>1</sup>

<sup>1</sup>Map via Google maps

## A-5 Pre-processing

For completion of this Appendix, Table A-4 and Table A-5 show the amount of lost and remaining data samples after pre-processing of Prinses Alexia, for respectively free and wake stream, as was also provided in Section 5-3.

**Table A-4:** Prinses Alexia turbines lost data-points due to filtering in free stream

Turbine Sector	R01 290° - 40° & 180° - 220°		R02 280° - 0° & 100° - 220°	
	Remaining points	Removed points	Remaining points	Removed points
Combining dataset	124272	-	124272	-
1. Alarms	121917	2355	122656	1616
2. Missing points	106086	15831	120275	2381
3. Error flags	98225	7861	113388	6887
4. Non-zero power values	79604	18621	90846	22542
5. Velocity limits	79366	238	90781	65
6. Minimum temperature (0°)	77254	2112	88607	2174
7. Wind direction boundaries	50953	26301	48983	39624
8. Time boundaries	50953	0	48983	0
9. Abnormal data filter	<b>44701 (35.97%)</b>	6252	<b>42445 (34.03%)</b>	6538

**Table A-5:** Prinses Alexia turbines lost data-points due to filtering in wake stream

Turbine Sector	R01 163° - 348°		R02 164° - 281° & 343° - 348°	
	Remaining points	Removed points	Remaining points	Removed points
Combining dataset	124272	-	124272	-
1. Alarms	121917	2355	122656	1616
2. Missing points	106086	15831	120275	2381
3. Error flags	98225	7861	113388	6887
4. Non-zero power values	79604	18621	90846	22542
5. Velocity limits	79366	238	90781	65
6. Minimum temperature (0°)	77254	2112	88607	2174
7. Wind direction boundaries	15412	61842	17783	70824
8. Time boundaries	15412	0	17783	0
9. Abnormal data filter	<b>13811 (11.11%)</b>	1601	<b>15915 (12.81%)</b>	1868

---

# Appendix B

---

## Wind farm Pen y Cymoedd

In this Appendix the site description, conditions and pre-process results for wind farm Pen y Cymoedd (PyC) are discussed. In Chapter 4 the focus was on Prinses Alexia as this wind farm is of primary interest due to the datasets size. However, PyC is used for the Business Case Refinement model and partly for some initial validation of the Power curve verification (PCV) model. While the focus of Chapter 4 is on Prinses Alexia, several aspects of PyC were discussed, which are repeated here to provide the full picture in this Appendix.

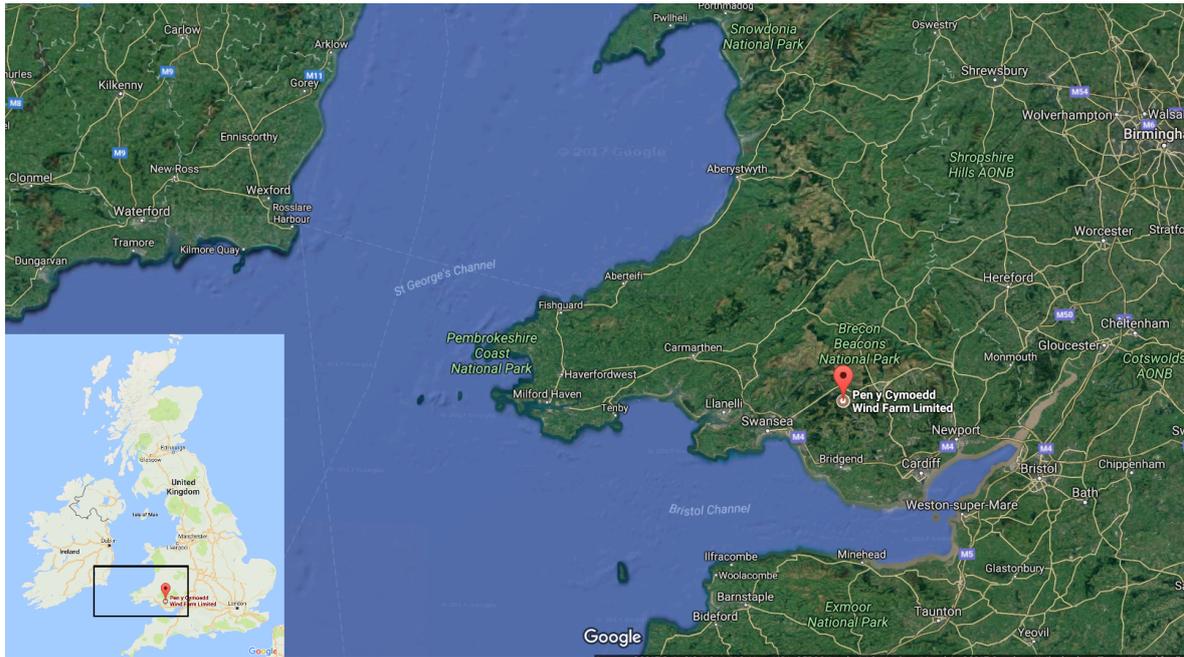
### B-1 Wind farm site

PyC is located near Cardiff in the south of Wales, United Kingdom, as location is shown in Figure B-1, and has been operational since 2016. The terrain is classified as complex terrain, due to the commercial forestry and hilly environment, with elevations ranging up to 600m. The turbines themselves are located between 280m and 550m above sea level.

On site, a total of 15 meteorological masts were installed, consisting of:

- 5 Development masts.
- 5 'Permanent' masts (from which 2 temporary), also known as reference masts.
- 5 Site calibration masts (replaced by turbines), also known as turbine masts.

The **development masts** were placed prior to development of the wind farm. These assess the site conditions for a period of time, from which, among others, the farm lay-out and business case are derived. Usually these development masts are removed quickly after the wind farm is realized, but in this case two remain until the end of 2018. **Permanent masts** are placed some time (in terms of months) prior to the construction of the wind farm turbines and are used for constant site condition monitoring and assessment, as well as for site calibration if calibration is necessary. Site calibration itself is treated in Chapter 4-2. The **site calibration mast** are replaced by wind turbines after a short period of time (usually a couple of months).



**Figure B-1:** Location of PyC. Map via Google Maps.

A detailed map of the locations of the turbines and meteorological masts, along with their installation and removal data can be found in Figure B-2. At the moment of this writing, the only turbines which had a met mast in the neighborhood were turbines T04, T13, T61, T82 and T83. However, PCV was only available for T82 and T83 and the data from this verification is used in this thesis.

From the development masts site specifications at hub height are determined using criteria from the Climatic condition review (CCR), as mentioned in the contract with the turbine manufacturer Siemens. [21] These specifications, shown in Table B-1, give an indication of the conditions on the site at hub height, in general.

**Table B-1:** Site specific conditions at hub height [21]

Site specific conditions	West	Central West	Central East	East
WTG quantity	11	19	33	20
WTG range	1-11	12-30	31-63	64-83
Air density [ $\text{kg}/\text{m}^3$ ]	1.193	1.186	1.175	1.185
Mean wind speed [m/s]	6.8	7.37	8.03	7.89
Weibull A [m/s]	7.67	8.32	9.07	8.97
Weibull k	2.04	2.27	2.12	2.07
TI @ 13m/s [%]	23%	23%	22%	24%
TI @ 14m/s [%]	22%	22%	21%	23%
TI @ 15m/s [%]	21%	21%	21%	21%
Inflow angle [°]	4.10	1.73	2.16	3.11
Wind shear [-]	0.29	0.23	0.27	0.33

In PyC two different turbines are present: Siemens SWT-3.0-108 and Siemens SWT-3.0-113, the details of these turbines are shown in Table B-2. The locations of all turbines and met masts is shown in Figure B-2. Today the day, there is a small wind farm located East of PyC. The location of this wind farm is shown in Figure B-3.

**Table B-2:** PyC turbine specifications. [23]

Turbine type	SWT-3.0-108	SWT-3.0-113
Turbines	T82, T83	T04, T13, T61
Rated power	3000 kW	3000 kW
Hub height	89.5m	88.0m
Rotor diameter	108m	113m
Rated wind speed	12.5 m/s	12.5 m/s
Cut-in wind speed	4.0 m/s	4.0 /s
Cut-out wind speed	25.0 m/s	25.0 m/s

## B-2 Measurement equipment

The two different turbines present in PyC each have a different hub height. For this reason the met masts in the farm are specified for the turbine(s) in the vicinity. This is primarily noticed in the altitudes of measurement instruments. The type of instrument and number of them is kept the same for each mast, for both calibration and permanent masts. The specific heights for the measurement instruments is given in Table B-3.

**Table B-3:** Measurement equipment altitudes for both types of reference and calibration masts [23]

Turbines	SWT-3.0-108 (T82-T83)				SWT-3.0-113 (T04-T13-T61)			
<b>Reference mast</b>	<b>m</b>	<b>m</b>	<b>m</b>	<b>m</b>	<b>m</b>	<b>m</b>	<b>m</b>	<b>m</b>
Cup anemometer	89.5	85.225	62.5	35.5	88.0	83.725	59.75	31.5
Wind vane	85.225	31.5	-	-	83.725	27.5	-	-
Ultrasonic 3D anemometer	82.509	-	-	-	81.008	-	-	-
Pressure sensor	83.3	-	-	-	81.8	-	-	-
Temperature/humidity sensor	83.3	-	-	-	81.8	-	-	-
<b>Calibration mast</b>	<b>m</b>	<b>m</b>	<b>m</b>	<b>m</b>	<b>m</b>	<b>m</b>	<b>m</b>	<b>m</b>
Cup anemometer	89.5	85.225	62.5	35.5	88.0	83.725	59.75	31.5
Wind vane	85.225	31.5	-	-	83.725	27.5	-	-
Ultrasonic 3D anemometer	-	-	-	-	81.008	-	-	-

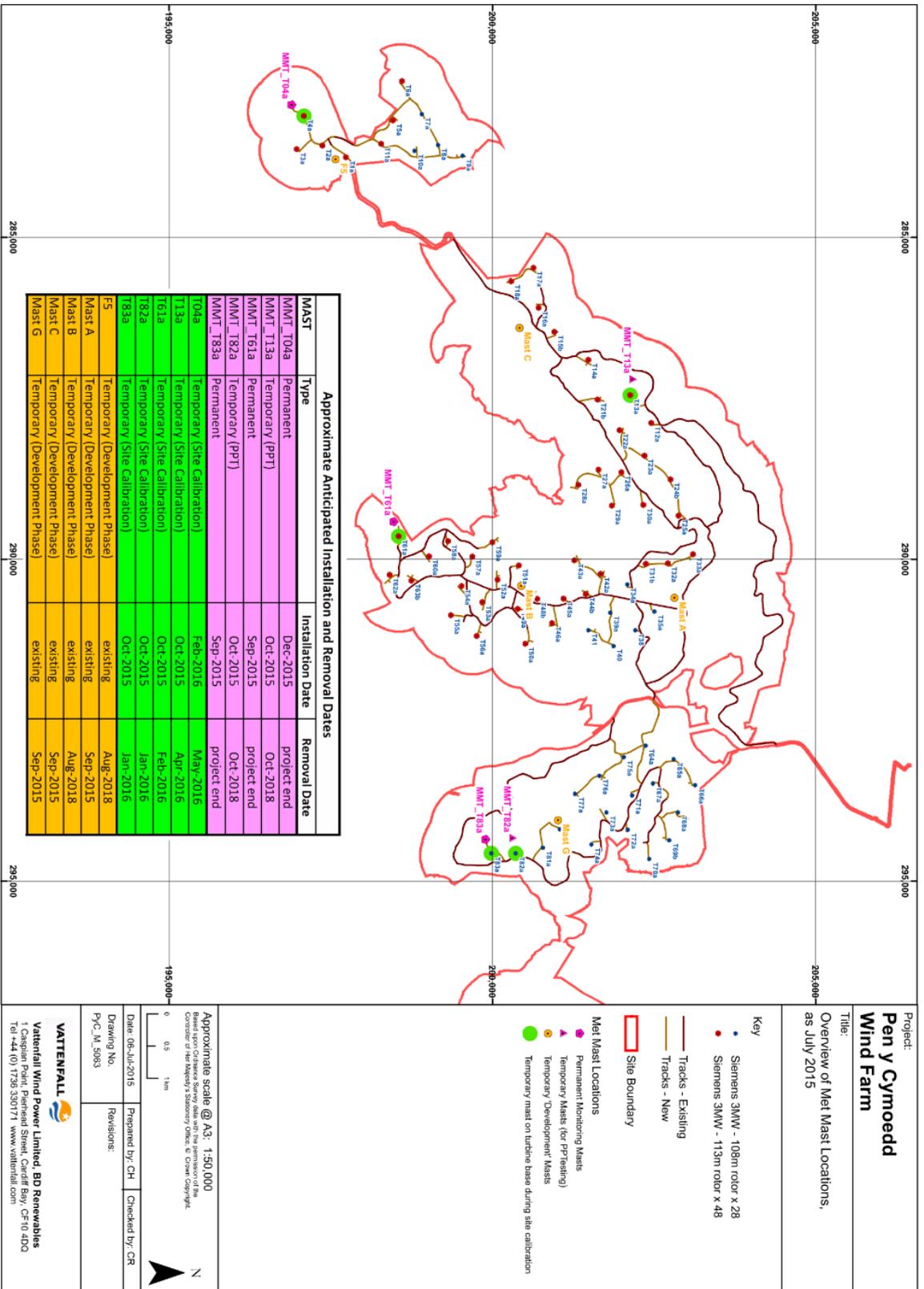


Figure B-2: Pyc windfarm location of meteorological masts and turbines [16]



**Figure B-3:** Additional wind turbines at the South-East border of PyC, map via Google maps

**Table B-4:** Wind velocity calibration relations. [24]

Turbine	Calibrated hub height velocity $U_{hub}$
T04	$0.98 \cdot U_{ref} + 0.09$
T13	$0.96 \cdot U_{ref} + 0.22$
T61	$U_{ref} + 0.082$
T82	$0.98 \cdot U_{ref} - 0.11$
T83	$0.97 \cdot U_{ref} + 0.13$

### B-3 Site calibration and power curve verification

The process of site calibration and power curve verification has been explained in Section 4-2. As was also mentioned there, site calibration has been performed at PyC due to the complex terrain. A previous MSc. thesis at Nuon, found linear relations for the site calibration of the five mentioned turbines. These are shown below in Table B-4.

Furthermore, the third party GL Garrad Hassan Deutschland GmbH (DNV-GL) also performed the PCV, of which data for turbines T82 and T83 is available at the moment of the writing of this thesis. With the current met masts and paired turbines, a total of 5 IEC approved power curve verifications can be performed. The other turbines are still waiting to be validated and are therefore not used in this thesis project.

## B-4 Measurement sectors

All wind directions are binned for three different sectors, depending on upwind conditions as discussed in Chapter 4: free stream, wake stream and blocked stream. The shift from wake to blocked sector if a mast is in the wake of a turbine has been determined at 4D by the IEC. [14]. Since both turbines T82 and T83 and the paired met mast are fairly close to each other the distances as well as the wake influence angle is evaluated for each combination. To define the sector for each wind direction bin, the distance from met mast to the paired turbine, but also the met mast of T82 to turbine T83 and vice versa, are used. The distance between these 4 locations is shown in Table B-5. Corresponding to these distances an angle is proved for which the wind direction is considered blocked. This is described in section 6.3.3 of the International Electrotechnical Commission (IEC) [14], as discussed in Section 4-4. Table B-5 also shows this corresponding angle behind the mast, which is taken as the limit of the blocked sector. Distances between mast and turbine which are not paired are shown in Figure B-4 and B-5. The sector for a distance of 2.5D is specifically stated in the IEC, while the sector for 3.8D is interpolated between 3D and 4D. Distances over 4D are considered wake, and therefore the wake from turbine T83 to the reference mast paired to T82 is not considered under blocked, but under wake.

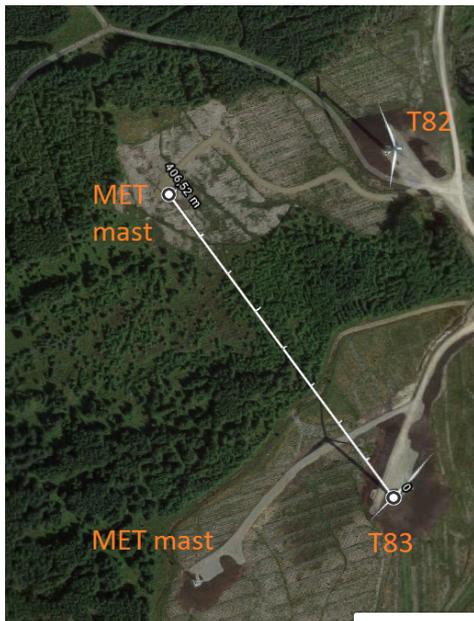
**Table B-5:** Distances and blocked sector between met masts and turbines

Turbine	met mast	Distance (x·diameter)	Blocked sector (° behind mast)
T82	Paired to T82	2.5D	74°
T82	Paired to T83	4.7D	Considered wake
T83	Paired to T82	3.8D	≈ 30°
T83	Paired to T83	2.5D	74°

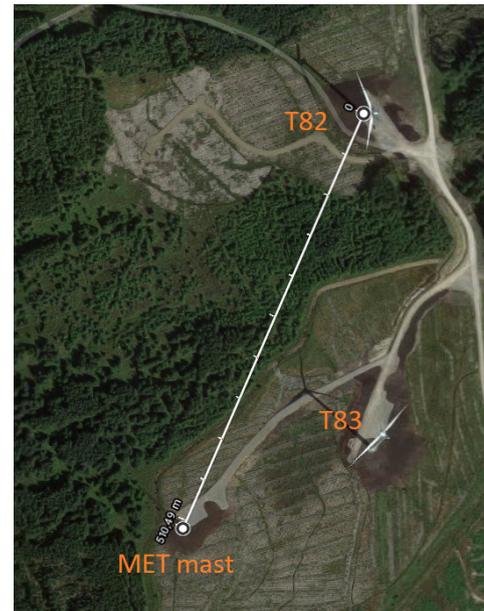
The remaining sectors, not considered free and not considered blocked, are placed in the wake sector. The resulting wind direction ranges for each sector are shown in Table B-6 and the sectors are shown in Figure B-6 and Figure B-7 for the met masts paired to T82 and T83 respectively.

**Table B-6:** Turbine wind sector ranges for PyC.

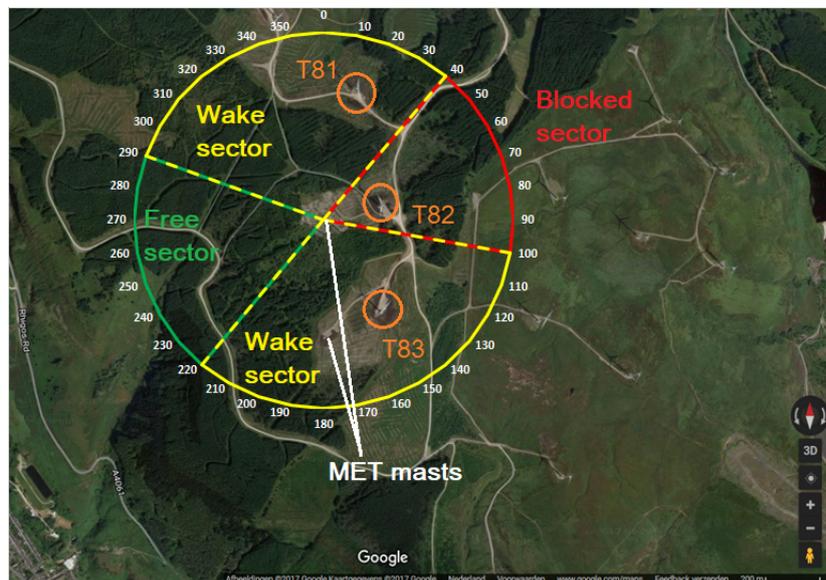
	Free sector	Wake sector	Blocked sector
T82	220° - 290°	290° - 40°	40° - 100°
	-	100° - 220°	-
T83	218.7° - 278.7°	280° - 0°	0° - 100°
	-	100° - 218.7°	-



**Figure B-4:** Distance of turbine T83 to met mast paired to turbine T82.<sup>1</sup>



**Figure B-5:** Distance of turbine T82 to met mast paired to turbine T83.<sup>1</sup>



**Figure B-6:** Wind direction sectors from reference mast paired to T82 <sup>1</sup>

<sup>1</sup>Map via Google maps

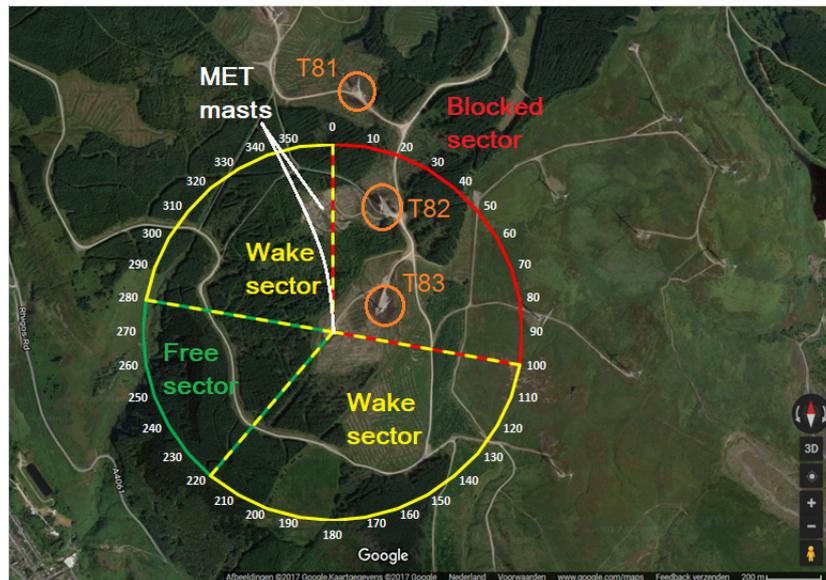


Figure B-7: Wind direction sectors from reference mast paired to T83 <sup>2</sup>

---

<sup>2</sup>Map via Google maps

## B-5 Pre-processing

In the same manner as for Prinses Alexia the data for PyC is pre-processed, the process being described in Chapter 5. A difference is that for PyC site calibration is performed and so the measured wind speed is slightly corrected. The resulting loss of data points and final dataset size for the free and wake sectors is shown in respectively Table B-7 and Table B-8.

**Table B-7:** Lost data-points due to filtering in free stream

Turbine Sector	T82 / UPYL05 220° - 290°		T83 / UPYL06 218.7° - 278.7°	
	Remaining points	Removed points	Remaining points	Removed points
Combining dataset	13092	-	33289	-
1. Alarms	9856	3236	28084	5205
2. Missing points	9856	0	28084	0
3. Error flags	8294	1562	20555	7529
4. Non-zero power values	8294	0	20555	0
5. Velocity limits	8089	205	20553	2
6. Minimum temperature (0°)	6816	1273	18959	1594
7. Wind direction boundaries	1577	5239	6505	12454
8. Time boundaries	1577	0	6505	0
9. Abnormal data filter	<b>1498</b> (11.4%)	79	<b>6201</b> (18.6%)	304

**Table B-8:** Lost data-points due to filtering in wake stream

Turbine Sector	T82 / UPYL05 290° - 40° & 180° - 220°		T83 / UPYL06 280° - 0° & 100° - 220°	
	Remaining points	Removed points	Remaining points	Removed points
Combining dataset	13092	-	33289	-
1. Alarms	9856	3236	28084	5205
2. Missing points	9856	0	28084	0
3. Error flags	8294	1562	20555	7529
4. Non-zero power values	8294	0	20555	0
5. Velocity limits	8089	205	20553	2
6. Minimum temperature (0°)	6816	1273	18959	1594
7. Wind direction boundaries	2342	4474	5175	13784
8. Time boundaries	2342	0	5175	0
9. Abnormal data filter	<b>2218</b> (16.9%)	124	<b>4940</b> (14.8%)	235



## Wind farm Norrekær Enge I

In this Appendix the site conditions of Norrekær Enge I (NKE) are discussed in more detail to provide a more complete picture of the wind farm. In Chapter 4 the wind farm was briefly mentioned, but due to the small size of the dataset this farm is not the main interest in this thesis project. Some features mentioned here were also mentioned in Chapter 4, but for completeness these are repeated in this Appendix.

### C-1 Wind farm site

Wind farm NKE is located in the North of Jutland, the main peninsula of Denmark, along a body of water West of Aalborg. Figure C-1 shows the location of the turbines and the wind farm itself. The terrain around NKE is comparable to Prinses Alexia as there are few obstacles and the terrain is flat, partly a body of water. The atmospheric conditions are also comparable as the mean wind speed is 7.87 m/s, the mean density 1.225 kg/m<sup>3</sup> and the mean Turbulence intensity (TI) 4.35%.

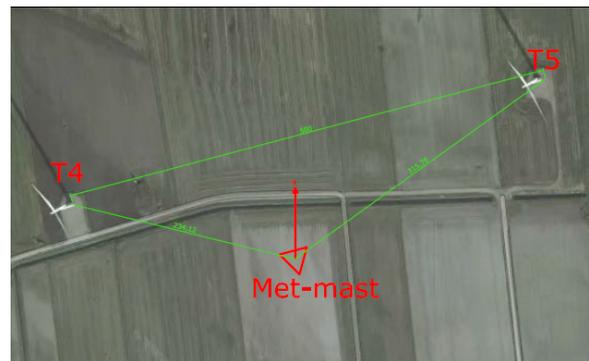
There are 13 turbines installed in NKE, all of which are of type Siemens SWT-2.3-93, some details about this turbine are shown in Table C-1. There is currently one operating meteorological mast, which was placed only later, in November 2014. This mast is in between turbines T04 and T05, as can be seen in Figure C-2. [17] This mast is paired to T04 and T05 only as measured conditions can be assumed similar at the met mast location as well as turbine location, for the undisturbed flow.

**Table C-1:** Siemens SWT-2.3-93 specifications

Parameter	Value	Parameter	Value
Hub height	80.0m	Rated wind speed	14.0 m/s
Rotor diameter	93.0m	Cut-in wind speed	4 m/s
Rated power	2300 kW	Cut-out wind speed	25.0 m/s



**Figure C-1:** Location of Norrekær Enge I, its turbines and the (later placed) meteorological mast. Maps via Google Maps.



**Figure C-2:** Location of the later placed met mast in NKE. [17]

## C-2 Measurement equipment

The met mast is placed in November 2014 at 234m from T04 and 315m from T05, as shown in Figure C-2. The measurement instruments located in the mast, together with their altitude, is shown in Table C-2.

## C-3 Power curve verification

Power curve verification (PCV) for NKE has been performed by Barlovento Recursos Naturales, but this dataset is not available. Furthermore, another met mast is used for this PCV analysis, a NASA MERRA node which is close to the wind farm. [22] For these reasons the PCV is not relevant to this thesis assignment.

The data which was available consisted out of two different datasets, one from the met mast and another containing the turbine Supervisory Control And Data Acquisition (SCADA)

**Table C-2:** Measurement equipment and altitude. Data via [17] and NKE datasets

Measurement instrument	Altitude above ground
Cup anemometer 1	80.0m
Cup anemometer 2	78.0m
Cup anemometer 3	57.0m
Cup anemometer 4	33.0m
Sonic spinner anemometer	76.0m
Wind vane 1	78.0m
Wind vane 2	57.0m
Wind vane 3	33.0m
Pressure sensor	76.0m
Temperature/humidity sensor	78.0m

data. After reviewing these, however, the timestamps in both sets were not the same 10-minute time-frames as there was a time-shift between these sets. This time-shift is not uncommon in combining mast data with SCADA data as these were measured by different instrument and are calibrated differently. Additionally the definition of the timestamp might be different; one may define this as the start of the 10-min averaged bin, while another might define this as the middle or end of the bin. To determine the correct time-shift the wind speed or temperature measured at the mast and as measured at the turbine hub are qualitatively compared. As the measurements at the turbine are influenced by the rotor (if operational), trends are not likely to align perfectly. However, trends as seen through peaks and valleys should be comparable and are therefore used to match data samples. Since wind speed is highly affected by the rotation of the rotor, while the temperature is assumed to be affected less, the temperature is used to determine the correct time-shift.

In this process a certain time-window is isolated and the temperature measured at the turbine hub and the mast are plotted for their corresponding time-stamp. Then by matching peaks and valleys of both trends the time-shift is determined. Figure C-3 shows the temperature measured at the turbine in black, the original temperature measured at the mast in green. The latter dataset is shifted by 60 minutes, shown as the red curve. For this time-shift the peaks and valleys match and this time-shift is therefore applied to the mast data.

## C-4 Measurement sectors

As discussed in Section 4-4, the wind directions (bins) are categorized in undisturbed upwind conditions, wake stream and blocked conditions. The latter includes cases where the mast is in the wake of the paired turbine and the distance is closer than  $4D$  as well as non-similar conditions, where either the turbine or the mast experiences disturbances (e.g. wake) but the other does not. For determining the relevant wind sectors the assessment made by the met mast contractor, DTU (Technical University of Denmark), as is shown in Figure C-4. [17] This yields in wind direction sectors as provided in Table C-3. These sectors are also shown in Figure C-5, with (rounded) wind direction bins.

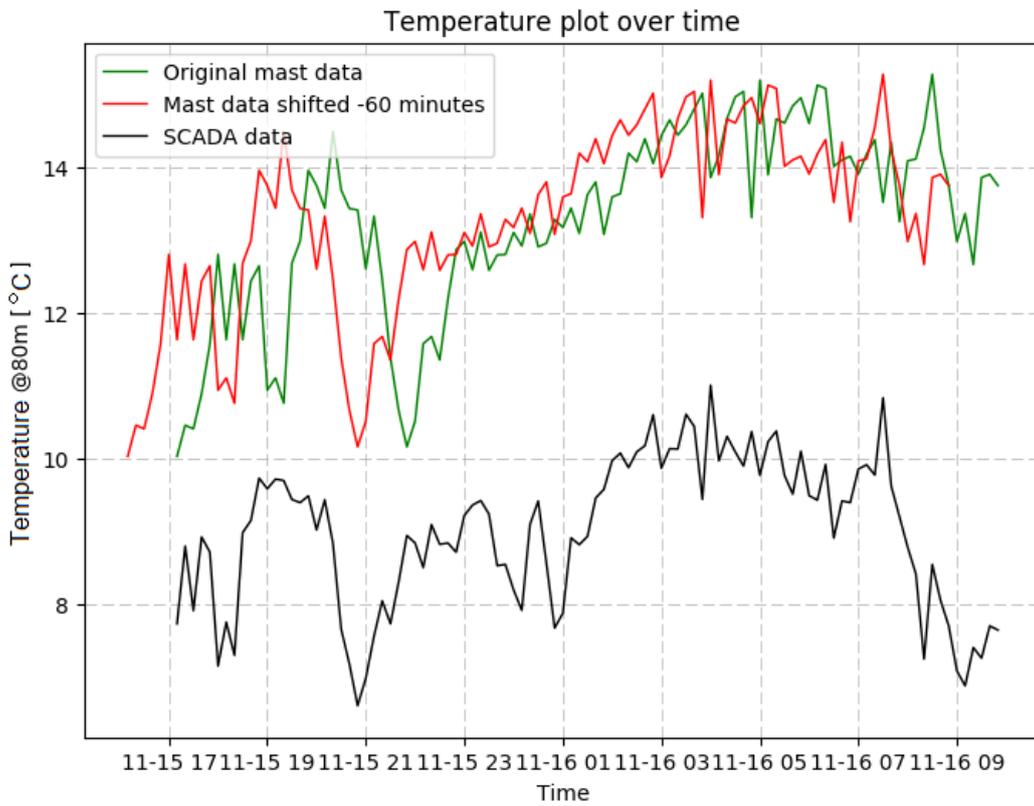


Figure C-3: 10-Minute averaged measured temperature at met mast and at the hub (from SCADA. The red line is the met mast shifted by -60 minutes.)

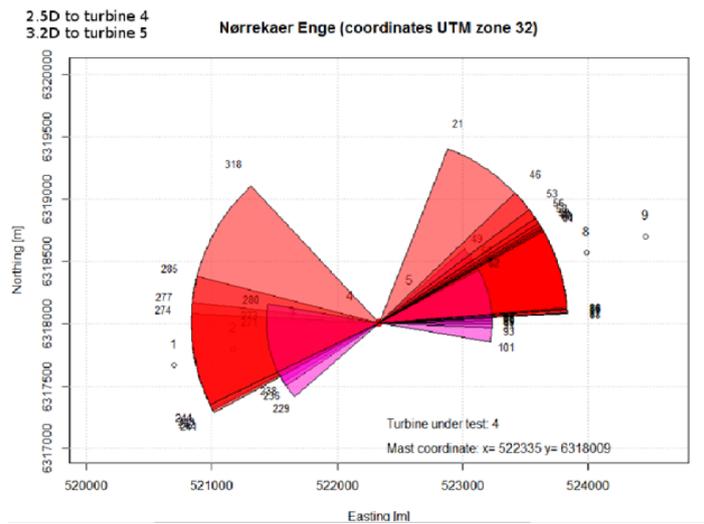
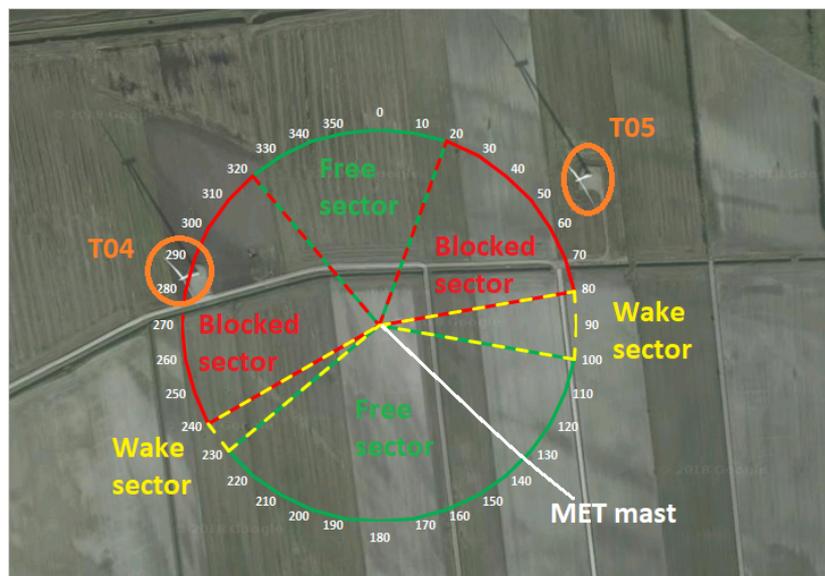


Figure C-4: Assessed influence directions of turbines on the meteorological mast.

**Table C-3:** Turbine wind sector ranges for NKE

	Free sector	Wake sector	Blocked sector
T04	101° - 229°	85° - 101°	21° - 85°
	318° - 21°	229° - 244°	244° - 318°
T05	101° - 229°	85° - 101°	21° - 85°
	318° - 21°	229° - 244°	244° - 318°

**Figure C-5:** Wind direction sectors for turbines T04 and T05, paired to the met mast. Map via Google Maps.

## C-5 Pre-processing

As described in Chapter 5 the initial dataset should be processed to be suitable as input of a neural network. The same procedure as elaborated upon there has been used on the data from NKE and the resulting data sample losses for the free and wake stream are shown in Table C-4 and Table C-5 respectively.

**Table C-4:** NKE turbines lost data-points due to filtering in free stream

Turbine Sector	T04 101° - 229° & 318° - 21°		T05 101° - 229° & 318° - 21°	
	Remaining points	Removed points	Remaining points	Removed points
Combining dataset	18281	-	18281	-
1. Alarms	18281	0	18281	0
2. Missing points	18281	0	18281	0
3. Error flags	11984	6297	11984	6297
4. Non-zero power values	10859	1125	10990	994
5. Velocity limits	10268	591	10373	617
6. Minimum temperature (0°)	7655	2613	7693	2680
7. Wind direction boundaries	3433	4222	3454	4239
8. Time boundaries	3433	0	3454	0
9. Abnormal data filter	<b>2958 (16.18%)</b>	475	<b>3055 (16.71%)</b>	399

**Table C-5:** NKE turbines lost data-points due to filtering in wake stream

Turbine Sector	T04 85° - 101° & 229° - 244°		T05 85° - 101° & 229° - 244°	
	Remaining points	Removed points	Remaining points	Removed points
Combining dataset	18281	-	18281	-
1. Alarms	18281	0	18281	0
2. Missing points	18281	0	18281	0
3. Error flags	11984	6297	11984	6297
4. Non-zero power values	10859	1125	10990	994
5. Velocity limits	10268	591	10373	617
6. Minimum temperature (0°)	7655	2613	7693	2680
7. Wind direction boundaries	1175	6480	1176	6517
8. Time boundaries	1175	0	1176	0
9. Abnormal data filter	<b>1027 (5.62%)</b>	148	<b>1019 (5.57%)</b>	157

---

# Appendix D

---

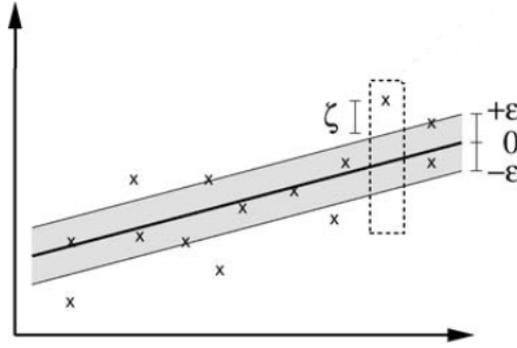
## Considered machine learning methods

In Chapter 3 the choice on the Neural network (NN) was elaborated. However, from the literature study several other interesting machine learning methods were considered for this thesis. [34] These are briefly discussed below.

### D-1 Support vector regression

A method which has also been compared in literature is the Support vector machine (SVM). While SVM can be used for both classification (not relevant to this thesis), regression (known as Support vector regression (SVR)) and more machine learning applications. SVM was invented by Vladimir Vapnik in 1992. [63] The basis idea is to find one or more hyperplanes which classify the provided data points and maximizes the margin between these classes. The larger the margin, the better the generalization. But this sounds like classification and not regression. However, the main features relevant for classification can also be used for regression analysis, as a non linear relation is learned/trained by using a linear learning machine mapping into a higher dimensional kernel induced feature space. This may seem very abstract, but will be explained later. An advantage in this method is that the hyper-parameters controlling the algorithm are independent of the input and output domain.

SVR is, like most machine learning algorithms, provided with a certain set of training data where a certain input pattern ( $x_i$ ) is paired to an output pattern ( $y_i$ ). The goal is then to find a function  $f(x)$  which represents the training data (the goal of most machine learning applications). In SVR, this is done by including an  $\varepsilon$ , which defines a 'tube' with radius  $\varepsilon$  around the function  $f(x)$  for which no error is allocated to the data points. The goal is then to optimize  $f(x)$  such that the  $\varepsilon$ -tube captures the most points from provided training targets,  $y_i$ , while at the same time remaining as flat as possible. This flatness is required to prevent over-fitting of the data. Figure D-1 shows a linear SVR, including the  $\varepsilon$ -tube. So the  $\varepsilon$  can be seen as a margin region around the function, where no error is applied to points (or errors  $\leq \varepsilon$  from the function are just ignored).



**Figure D-1:** Linear SVM, the gray area indicates the  $\varepsilon$  'tube' [18]

Written in a function (in this case for the linear example), Eq. (D-1) gives the form of the regression function.  $\chi$  denotes the domain of input parameters (for example  $\chi = \mathbb{R}^d$ , with  $d$  the amount of input parameters). By minimizing  $w$  the flatness of  $f(x)$  is increased. To do this an optimization form can be expressed (Eq. (D-2)). What  $w$  will consist off will be clear later. Here the norm  $\|w\|^2 = \langle w, w \rangle$  is minimized.  $\langle w, w \rangle$  means the dot product of  $w$  with  $w$ , in the domain  $\chi$ .

$$f(x) = \langle w, x \rangle + b \text{ with } w \in \chi, b \in \mathbb{R} \quad (\text{D-1})$$

$$\begin{aligned} & \text{minimize} && \frac{1}{2} \|w\|^2 \\ & \text{subject to} && \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon \end{cases} \end{aligned} \quad (\text{D-2})$$

There is a catch, however. The above method assumes the optimal function  $f(x)$  exists with a precision, defined via  $\varepsilon$ , and that this can be optimized (through equation (D-2)). But one might encounter problems when this does not hold, resulting in an infeasible optimizing problem. To solve this, some errors are introduced via the so-called *soft margin* loss function. This method concerns the introduction of slack variables  $\xi_i$  and  $\xi_i^*$ . This allows to solve the infeasible constraints, mentioned earlier. In optimization form, Eq. (D-2) is extended to Eq. (D-3) and the function to minimize is called the empirical risk function,  $R_{emp}(x, \xi, \xi^*)$ . The parameter  $\ell$  represents the number points outside the  $\varepsilon$ -band. Constant  $C > 0$  represents the trade-off between flatness ( $\|w\|^2$  part) and the error deviations allowed ( $C \sum_{i=1}^{\ell} (\xi_i + \xi_i^*)$  part) and is called the penalty or regularization parameter. These slack variables also result in an  $\varepsilon$ -insensitive loss function  $|\xi|_{\varepsilon}$ , as shown in Eq. (D-4).

$$\begin{aligned} & \text{minimize} && R_{emp} = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{\ell} (\xi_i + \xi_i^*) \\ & \text{subject to} && \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned} \quad (\text{D-3})$$

$$|\xi|_\varepsilon := \begin{cases} 0 & \text{if } |\xi| \leq \varepsilon \\ |\xi| - \varepsilon & \text{otherwise} \end{cases} \quad (\text{D-4})$$

Now Figure D-1 can be extended with a soft margin loss or  $\varepsilon$ -insensitivity. Each point outside the margin has a contribution to the cost function (which has to be minimized), as shown in Figure D-2.

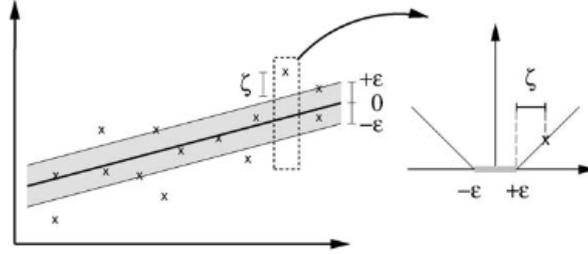


Figure D-2: Soft margin loss setting for a linear SVM, via [18]

### The dualized problem

Most problems, especially non-linear ones, can in general be solved more easily by using dual formulation. [18] To solve dualization, Lagrange multipliers are used. For this, first the objective function has to be setup as a Lagrange function (called the primal objective function), as well as its corresponding constraints. This is done by setting a dual set of variables. For example, the previous example (Eq. (D-3)) can be written in said Lagrange form as follows (Eq. (D-5)):

$$\begin{aligned} \mathcal{L} := & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{\ell} (\xi_i + \xi_i^*) - \sum_{i=1}^{\ell} (\eta_i \xi_i + \eta_i^* \xi_i^*) - \\ & \sum_{i=1}^{\ell} \alpha_i (\varepsilon + \xi_i - y_i + \langle w, x_i \rangle + b) - \\ & \sum_{i=1}^{\ell} \alpha_i^* (\varepsilon + \xi_i^* + y_i - \langle w, x_i \rangle - b) \end{aligned} \quad (\text{D-5})$$

Here  $\mathcal{L}$  is the Lagrange and the Lagrange multipliers are introduced as  $\alpha_i$ ,  $\alpha_i^*$ ,  $\eta_i$  and  $\eta_i^*$ , which are all positive. Then by taking the partial derivatives with respect to the so-called primal variables (being  $w$ ,  $b$ ,  $\xi_i$  and  $\xi_i^*$ ) and setting them to 0 yields in the optimal point from the saddle point condition (Eq. (D-5)).

$$\partial_b \mathcal{L} = \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) = 0 \quad (\text{D-6})$$

$$\partial_w \mathcal{L} = w - \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) x_i = 0 \quad (\text{D-7})$$

$$\partial_{\xi_i^{(*)}} \mathcal{L} = C - \alpha_i^{(*)} - \eta_i^{(*)} = 0 \quad (\text{D-8})$$

This dual formulation is then obtained by implementing Equations (D-6) to (D-8) in optimizing Eq. (D-5), yielding in Eq. (D-9). Note that some Lagrange multipliers ( $\eta_i$  and  $\eta_i^*$ ) are eliminated in the process as these are defined in Eq. (D-8). Combining this definition with Eq. (D-7), results in a definition of the so-called *Support Vector expansion*, shown in Eq. (D-10).

$$\begin{aligned} & \text{maximize} && \begin{cases} \frac{1}{2} \sum_{i,j=1}^{\ell} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle x_i, x_j \rangle \\ -\varepsilon \sum_{i=1}^{\ell} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{\ell} y_i (\alpha_i - \alpha_i^*) \end{cases} \\ & \text{subject to} && \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C] \end{aligned} \quad (\text{D-9})$$

$$w = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) x_i \text{ therefore } f(x) = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*) \langle x_i, x \rangle + b \quad (\text{D-10})$$

The latter part of Eq. (D-10) represents the solution for the optimization problem (Eq. (D-3)), this is the SVR regression function. Here  $\ell$  is the number of Support Vectors and the kernel function  $\langle x_i, x \rangle$  can be rewritten as Eq. (D-11), where  $g_j(x)$  represents a set of nonlinear transformations.

$$\langle x, x_i \rangle = \sum_{j=1}^m g_j(x) g_j(x_i) \quad (\text{D-11})$$

SVR aims to minimize the empirical risk  $R$ , which is shown in Eq. (D-12), which uses the  $\varepsilon$ -insensitivity function  $|\xi|_{\varepsilon}$ .

$$R_{emp}(f) = \frac{1}{\ell} \sum_{i=1}^{\ell} |\xi|_{\varepsilon}(y_i, (\alpha_i - \alpha_i^*) \langle x_i, x \rangle + b) \quad (\text{D-12})$$

Application of the SVM in order to find a generalized function highly depends on setting the meta-parameters,  $C$ ,  $\varepsilon$  and the kernel parameters, comparable to the choice of hyper-parameters and functions in the neural network. These meta-parameters influence the complexity, and therefore the generalization (overly complex functions tend to be less generalized). The choice for meta-parameters is typically dependent on the problem, application domain and data set quality and size. SVM can generally find complex functions from relative small datasets. If the training time allows, a grid search can be used to find the optimal parameters. The main parameters to set are:

**Regularization parameter  $C$**  The  $C$  parameter determines, as mentioned before, the trade-off between complexity and the degree points outside the  $\varepsilon$  band are taken into account and therefore indicates tolerance of 'outliers'. This is shown in the objective function in Eq. (D-3).

**Insensitivity parameter  $\varepsilon$**  As mentioned earlier,  $\varepsilon$  determined the width of the ' $\varepsilon$ -tunnel/band' which defined the points without error, the  $\varepsilon$ -insensitivity zone. Large values of  $\varepsilon$  mean a wider tunnel and less support vectors. But small values of  $\varepsilon$  also mean more complex functions (as large values, on the other hand, tend to more flat results). Model complexity is therefore a combination of both  $C$  and  $\varepsilon$ .

**Kernels** Kernel functions are similarity or weighting functions, provided to a machine learning algorithm to basically provide for a shortcut to the mapping of the function to the training data. Certain calculations go a lot faster using kernels, especially if these calculations would involve the higher domain space (which is likely to happen for multivariate input). In support vector regression these kernel functions give a certain weight to the points deviating from the  $\varepsilon$ -tube. Points closer to the tube -where  $|x_i - x|$  is close to 0- have a large weight, and points far away receive a small weight therefore reducing their influence on the regression.

There are several kernel function types, just like there is a variation activation functions in neural networks. Most widely used are the Gaussian Radial basis functions (RBF) Eq. (D-13), with a width  $\sigma$  (scale parameter), and the polynomial kernel with order  $k$  and slope constant  $\alpha$  and constant  $c$ , shown in Eq. (D-14). [64] There are others, but these are barely used. From the two given kernel functions, the RBF is most easily implemented and capable of non-linear mapping of the training data. A visual comparison between RBF, polynomial and linear function are shown in Figure D-3. The non-linearity of the RBF is clearly an advantage in non-linear and non-polynomial relations.

$$\langle x_i, x \rangle = K(x_i, x) = e^{-\frac{\|x_i - x\|^2}{2\sigma^2}} \quad (\text{D-13})$$

$$\langle x_i, x \rangle = K(x_i, x) = (\alpha x_i x + c)^k \quad (\text{D-14})$$

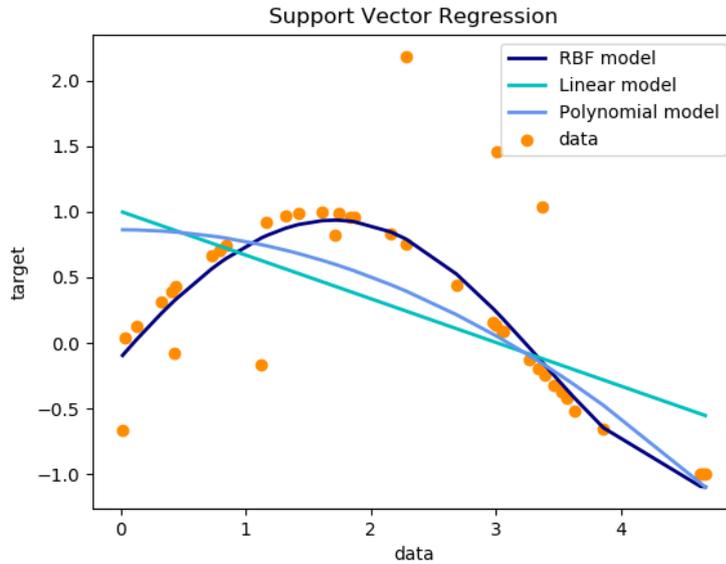
## D-2 Fuzzy logic

Another sort of optimization is based on fuzzy rules, which uses fuzzy if-then rules. These fuzzy rules are no absolute rules, but conditional statements and more or less similar to human logic. Instead of a 'hard' benchmark for decision making (e.g. IF  $(x \leq 1)$  THEN ...), the statements are labels or fuzzy sets which are characterized in so called membership functions (MF). This allows for imprecise modes of reasoning, which can be found in human reasoning. The principles of fuzzy logic is among other described by Jang. [19] An example of a fuzzy statement is:

```
1 IF pressure is high THEN volume is small
```

Here the labels (pressure and volume) are linguistic variables and the benchmarks (high and small) are linguistic values or labels. These values or labels are characterized in the membership functions (MF). Another example is partly use of fuzzy logic, only in the premise part. This is proposed by Takagi and Sugeno, and takes the following form: [65]

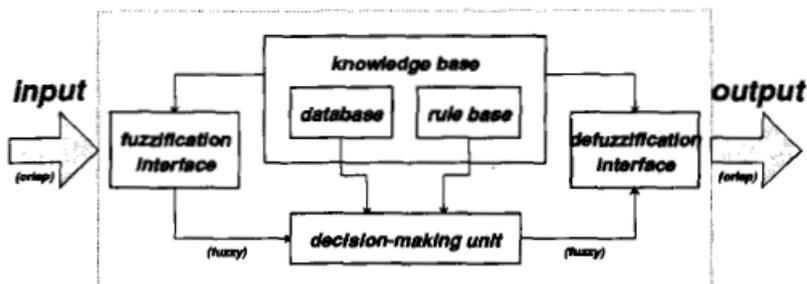
```
1 IF pressure is high THEN F = kv^2
```



**Figure D-3:** Comparison of fitting capabilities to training data of linear, polynomial and RBF kernel functions. (via Python Scikit package examples)

Here only the premise is linguistic and therefore characterized by a membership function. The use of fuzzy logic has been adapted in modeling and control, where it is able to capture the idea of a 'rule-of-thumb'.

A fuzzy logic system also works from input and certain output. But instead of using some function to be determined, it is rule based. An overview of a fuzzy inference system is shown in Figure D-4. The system consists of a rule base (containing fuzzy IF-THEN rules), a database (defines the membership functions), a decision making unit (performs inference operation on rules), 'fuzzification' interface (transform input to fuzzy values and labels) and a 'defuzzification' interface (transforms fuzzy results back to required output type).



**Figure D-4:** Overview of fuzzy inference system [19]

A fuzzy system works by first transforming the input to fuzzy values, parameters and labels. The input is then compared to membership functions in the premise (IF-part) to obtain fuzzy membership values. Then the premise membership values are combined which yield in a so called *firing strength*, or weight, of each rule from the database. Depending on this firing

strength the consequence is determined for each rule. All consequences are then 'defuzzified' and the system produces an output.

While fuzzy logic is an interesting optimization solver, there is a large complication with this method. It is rule-based and these rules have to be set-up upfront. These rules then use the membership functions to determine the output, but these membership functions are highly subjective. This makes sense, since fuzzy logic tries to approach human reasoning which is highly subjective. But this also means objective (or 'crisp') outcome is not very likely and this will likely result in a rather coarse approximation of the problem. Tuning of a Fuzzy inference system (FIS) is therefore usually done by an expert, which limits the applicability of the system. [20] A comparison between artificial neural networks and fuzzy inference systems has been made by Abraham. [26] Table D-1 gives a brief overview of the advantages, disadvantages and/or limitations of both systems. It becomes clear fuzzy inference systems rely much in prior knowledge, which is quite the limitation in the thesis assignment.

**Table D-1:** Comparison between neural network and fuzzy logic systems [26]

Artificial neural network	Fuzzy inference system
Difficult to use a priori knowledge	Prior rule-based can be incorporated
Learning from scratch	Cannot learn (linguistic knowledge)
Black box	Interpretable (if-then)
Complicated learning algorithms	Simple interpretation and implementation
Difficult to extract knowledge	Knowledge must be available

## Neuro-fuzzy systems

So FIS are not suitable for the task ahead. But what about a novel technique of combining a neural network with fuzzy logic to use the learning property of the ANN with the analyzing capabilities of FIS? A relative new method is the use of Adaptive neuro-fuzzy interference system (ANFIS), also known as adaptive-network-fuzzy interference system, and can be seen as a combination of a neural network and fuzzy logic. ANFIS models can learn non-linear signal relations by using the fuzzy logic described above, using Membership function (MF) parameters in the training phase. Since there usually are fewer membership parameters than neuron parameters to train, ANFIS is usually quicker in training. Furthermore, some a priori knowledge can be incorporated in the tunable rules and membership functions, while neural networks work as a black box. [20]

There are a number of different methods of modeling with combining neural network, fuzzy logic and even evolutionary programming together presented by Abrahams. [26] Within these methods there is a distinction of cooperative NN/FISIS, concurrent NN/FIS and integrated NN/FIS systems, increasing in complexity and advantages. One method mentioned earlier is ANFIS, an integrated NN/FIS system. In its simplest form ANFIS works in several stages. The NN is used as a pre-processor to train the FIS parameters from the training data in a fixed algorithms structure. The adapted MFs and rules are then used in the model to predict model output. However, initial membership functions and thus a priori knowledge or assumptions is still required. [19]

## Comments on FIS

Commenting on the potential of FIS on the thesis assignment, FIS is not suitable due to the required upfront definition of the membership functions. Also the fact that the dependency between input and output is highly subjective and should be tuned by an expert makes this method unattractive. A combination of a neural network and FIS proves to be more interesting. Training the membership functions can yield in a model tuned by training such models are highly flexible and can provide with more insight than the black box neural network. However, prior knowledge is still required in setting up membership functions and rules. Furthermore, while the theory on these models is available, although limited, there are currently almost no reliable models available which incorporate this kind of modeling. For the thesis assignment this would mean programming of such a model from scratch which is likely to take too long and is beyond the skills of the author. For this reason these kinds of algorithms are not realistic for now, despite their promising use.

## D-3 Regression/decision trees

A different approach, although slightly similar to fuzzy logic, is the use of regression trees. Breiman et al. elaborate on classification and regression trees quite extensively. [66] The term regression indicates the predicted outcome is considered real value. In regression tree modeling, IF-THEN rules split the dataset into a number of subsets based on (a) certain feature(s) and related value. Thus in this data-mining application data is split, categorized and/or generalized. This method could potentially lead to a large number of levels with a large number of tree nodes per level, but this increases the risk of over-fitting. These rules have to be set up upfront, however, and this requires knowledge of the problem beforehand. The predictive capabilities then comes from going down (through) the regression tree, constructed from the dataset, and taking the average value of the remaining samples in the data subset. This process of splitting/regression is performed as optimal as possible to make the tree as optimal as possible. A disadvantage is that the tree can become extensive to such an extent that there is no learning process, it will not recognize a pattern from the dataset. In this case it only memorizes the data and this yields in over-fitting of the data. Several methods are available to improve the process of splitting the data and some of these are evaluated in literature, as pointed out in Chapter 2. The best evaluated method was Stochastic gradient boosted regression trees (SGBRT).

Janssens et al. mention a useful feature of using regression trees from multivariate input. [2] The relative importance of features in the dataset can be analyzed by going through the tree. Going from top to bottom, computing the error reduction at every node, multiplying this with the sample size through this node and adding this together for individual features yield, after normalization, in relative importance of these features.

Advantages of decision tree methodology are described by James et al. and include the simplicity of the model, use of a large number of variables, white box model approach, performance in large datasets and non-statistical approach (no assumption on prediction). [67] However, there are of course also some limitations. Decision tree models tend not to be as accurate as other data mining approaches, are prone to over-fitting or over-complexing trees

(and therefore no generalizing form) and finding of a global optimum over a local optimum is not guaranteed since optimum-based decisions are made at all nodes.

### **Stochastic gradient boosted regression trees**

SGBRT is a method of setting up (individual) regression trees using a 'forward stage-wise' procedure, which tests the addition of variables with a criterion. The statistically most significant improvement is added to the tree, step by step. Initially a single regression tree is trained from the data set. Using gradient boosting the initial tree is improved and subsequent models are trained from the error(s) of each previous model. Using a learning rate the contribution of an additional tree is scaled, limiting the over-fitting tendency of the regression tree method. Furthermore additional trees can be set up using a random (stochastic) selection of data samples, reducing the over-fitting risk and speeding up the process. [2]



## Neural network function influences

For constructing a neural network some choices have to be made; the activation function for each layer, the training/optimizing function and normalization functions have to be set prior to training. While the choice for activation, optimization and normalization functions were given, it is interesting to see how this changes the outcome of constructed model. To investigate this influence the constructed neural network is trained with the same dataset and hyper-parameters, but with changing optimization and normalization function. The results from this comparative study are given in this appendix.

### E-1 Optimization function

The function of the training, or optimization, function is to let the machine/computer optimize model internal unknowns from the provided training dataset. Depending on the dataset, some types of training functions are more suitable than others in terms of speed or getting stuck on local minima or saddle points. In Keras a number of optimizing functions are available, which will be tested and compared [27]:

- Stochastic gradient descent (SGD)
- Root Mean Square propagation (RMSprop)
- Adaptive Gradient algorithm (Adagrad)
- Adadelata
- Adaptive moment estimation (Adam)
- Adamax
- Nesterov-accelerated adaptive moment estimation (Nadam)

Some of these have been briefly discussed in Chapter 3-2. The others will not be explained in detail, but if the reader would want to know more about how these optimizers work, an excellent blog-post about stochastic gradient descent optimizers is advised [68].

To compare said optimizers, the training process is run 10-fold (ten times) with the same network hyper-parameters and the average accuracy of the optimizing function and average computational time are monitored. The model is trained for 75% of the full Prinses Alexia dataset and evaluated with the remaining 25%. For each test the training and testing datasets are constant. The model should therefore be more or less equal every time, otherwise the optimizer is not able to find the global optimum but is likely to get stuck in a local optimum. Evaluation is done by determining the Mean absolute error (MAE) of the difference between modeled and measured power for the testing dataset. The learning rate of each function, if applicable, is set constant at  $1e^{-4}$ , other optimizing function parameters are kept as default. The settings for the model are shown in Table E-1. Quantification of the accuracy (and more importantly capability of finding the global minimum for each run) is quantified with the standard deviation of the collected MAE. This should be as close to 0 as possible, as this will indicate less deviation between the solutions. The results for each run and the average time and standard deviations are shown in Table E-2.

The tests were performed on a laptop with Intel Core i5-7300HQ, 2.5 GHz over 4 cores, 8 GB RAM and two GPU's: Intel HD Graphics 630 4GB and NVIDIA GeForce GTX 1050 6GB.

**Table E-1:** Settings for each run in training function evaluation.

Parameter	Value
Layout (nodes/layer)	9-14-8-1
Epochs	5000
Batch size	512
Early stopping	40 epochs
Dropout rate	0.0
Regularization parameter	0.0
Learning rate	$1e^{-4}$

**Table E-2:** Average computational time and optimizer accuracy, trained for Prinses Alexia and averaged for 10 runs.

Training function	Average time [mm:ss]	Average MAE	Standard deviation MAE
SGD	12:01	71.9 kW	0.21 kW
RMSprop	02:27	70.3 kW	0.34 kW
Adagrad	12:23	71.7 kW	0.23 kW
Adadelat	12:25	72.4 kW	0.31 kW
Adam	04:28	70.0 kW	0.31 kW
Adamax	06:39	70.5 kW	0.44 kW
Nadam	05:41	70.2 kW	0.32 kW

From the table a number of conclusions can be deduced. First of all, the mean MAE of all optimizers is very close to each other, indicating they are all able to find a good solution for these settings. Furthermore it seems SGD, Adagrad and Adadelat all take at least two

times longer than the other optimizers. While this is not an issue in short, single simulation, for longer (or more) training processes this might become a problem. Finally the standard deviation of the MAE of all the optimizers are also fairly close to each other, with SGD having a minimum of 0.19 kW and the maximum value, Adamax, of 0.44 kW. From this table both RMSprop and Adam seem to be good choices taking into account computational time and accuracy.

## E-2 Activation function

In the constructed neural network the Exponential linear unit (ELU) activation function was chosen for the input and hidden layers and TanH for the output layer. While the choice for this function was argued in Chapter 3, there are several activation functions available. The domains of these functions is shown below (Table E-3). The available activation functions in Keras are described below. The most popular activation functions are Sigmoid, Tanh, Rectified linear unit (ReLU), ELU, Gaussian and Softsign. These are respectively plotted in Figure E-1 to Figure E-6.

Name	Range
ELU	$(-\alpha, \infty)$
Scaled exponential linear unit (SELU)	$(-\infty, \infty)$
Softplus	$(0, \infty)$
ReLU	$(-\infty, \infty)$
TanH	$(-1, 1)$
sigmoid	$(0, 1)$
Linear	$(-\infty, \infty)$
Softmax	$(0, 1)$
Parametric rectified linear unit (PReLU)	$(-\infty, \infty)$
Leaky ReLU	$(-\infty, \infty)$
Softsign	$(-1, 1)$

**Table E-3:** Keras activation functions and range [27]

### ELU

$$f(h) = \begin{cases} \alpha (e^h - 1), & \text{if } h < 0 \\ h, & \text{if } h \geq 0 \end{cases}$$

### SELU

$$f(h) = \begin{cases} \alpha (e^h - 1), & \text{if } h < 0 \\ h, & \text{if } h \geq 0 \end{cases}$$

with  $\lambda = 1.0507$  and  $\alpha = 1.67326$

### Softplus

$$f(h) = \ln(1 + e^h)$$

**ReLU**

$$f(h) = \begin{cases} 0, & \text{if } h < 0 \\ h, & \text{if } h \geq 0 \end{cases}$$

**TanH**

$$f(h) = \tanh(h) = \frac{2}{1 + e^{-2h}} - 1$$

**Sigmoid**

$$f(h) = \frac{1}{1 + e^{-h}}$$

**Linear/identity**

$$f(h) = h$$

**Softmax**

$$f_i(h) = \frac{e^{h_j}}{\sum_{j=1}^J e^{h_j}}$$

for  $i = 1, \dots, J$ .

Softmax is , however, not a function of a single fold  $x$  coming from the 'previous' layer.

**PReLU**

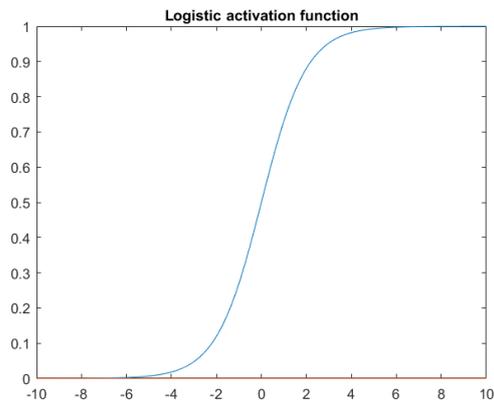
$$f(h) = \begin{cases} \alpha h, & \text{if } h < 0 \\ h, & \text{if } h \geq 0 \end{cases}$$

**Leaky ReLU**

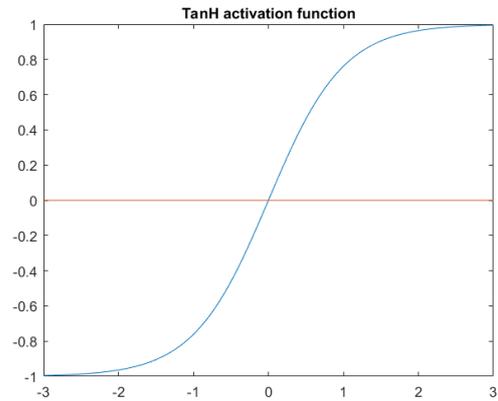
$$f(h) = \begin{cases} 0.01h, & \text{if } h < 0 \\ h, & \text{if } h \geq 0 \end{cases}$$

**Softsign**

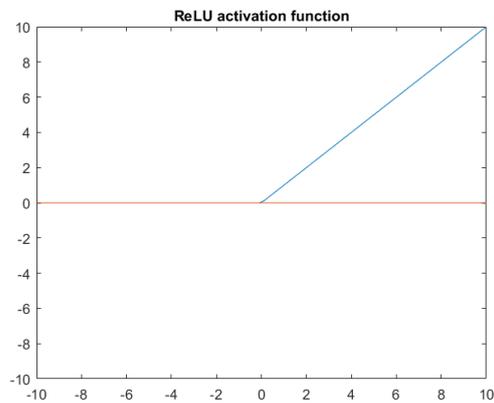
$$f_i(h) = \frac{h}{1 + |h|}$$



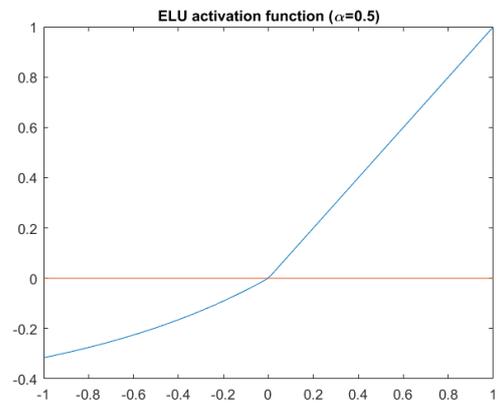
**Figure E-1:** Logistic/sigmoid activation function



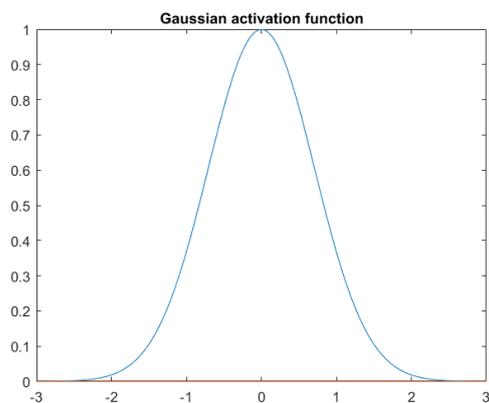
**Figure E-2:** Tanh activation function



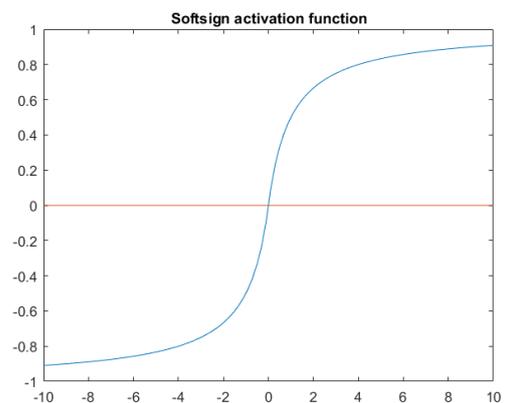
**Figure E-3:** ReLU activation function



**Figure E-4:** ELU activation function



**Figure E-5:** Gaussian activation function



**Figure E-6:** Softsign activation function

### E-3 Normalization scheme

In the end of Chapter 5 the normalization scheme of the output was discussed and chosen, which was the difference between measured/modeled and warranted power normalized to rated power, shown below in Eq. (E-1). This way more information about the shape of the power curve was contained, while maintaining a scalable output parameter for the turbine size. This would improve the ability for extrapolation to other wind turbines, but also flatten influences in general and result in a more generalized output. However, a comparison with the normalization of the performance to the rated power was also considered, but deemed to lose information about the Power curve (PC) shape. Therefore a comparison between both normalization schemes is performed. This comparison is only done for the Power curve verification (PCV) model, for wind farm Prinses Alexia, as this yields in sufficient data to train a model with sufficient complexity. If the model complexity is not sufficient (which is limited for small datasets) the comparison is not accurate and likely to be a bad representation of the normalization scheme.

#### Normalization scheme testing set-up

The used normalization is again shown in Eq. (E-1), while the normalization purely to rated power (e.g. dimensionless power curve) is shown in Eq. (E-2).

$$y_{train} = \frac{P_{measured} - P_{warranted}}{P_{rated}} = \frac{P_{measured}}{P_{rated}} - \frac{P_{warranted}}{P_{rated}} \quad (\text{E-1})$$

$$y_{train} = \frac{P_{measured}}{P_{rated}} \quad (\text{E-2})$$

To assess the performance of both schemes, the target data is reshaped with the above equations. Then the model is trained with the training input data (random 75% of the full dataset) and evaluated with the test set (the remaining 25% of the full test set). To validate the model itself, the modeled power curve is compared to the measured performance. This is done by plotting the difference from the warranted power curve, as the actual power-velocity of each sample just shows as a collection of samples following the power curve. No actual trends can then be seen besides overall -typical- power curve shape. By showing the difference from the warranted power curve, more trends are visible and can be compared for model validation. To quantify the performance of the model the MAE and Root mean squared error (RMSE) of the difference from the measurements (the training data) and R<sup>2</sup>-score of measured to modeled power are calculated for the test set. These are respectively shown below in Equations E-3, E-4 and E-5, which were introduced in Section 6-3.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (\text{E-3})$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (\text{E-4})$$

$$R^2 \equiv 1 - \frac{SS_{res}}{SS_{tot}} \quad (\text{E-5})$$

$$SS_{res} = \sum_{i=1} (y_i - \hat{y}_i)^2 \quad (\text{E-6})$$

$$SS_{tot} = \sum_{i=1} \left( y_i - \frac{1}{N} \sum_{i=1}^N y_i \right) \quad (\text{E-7})$$

The resulting MAE, RMSE and  $R^2$ -score of each method are compared to each other as well as to the warranted power curve method. This will show which method is most suited for this application. The network settings are almost the same as the initial grid search settings, presented in Section 6-3. These are repeated in Table E-4. The difference is in the stopping epoch amount, which is set at 50, while the maximum amount of epochs is sufficiently large to never reach this amount. The network lay-out is taken the same as the found PCV model, and the batch-size is halved. The range of network output differs for each method; normalization to rated power ranges from 0 to over 1, while normalization to difference from the warranted power ranges between -1 and 1. Therefore the output layer activation function is set at ELU and Tanh respectively.

**Table E-4:** Normalization scheme comparing network settings

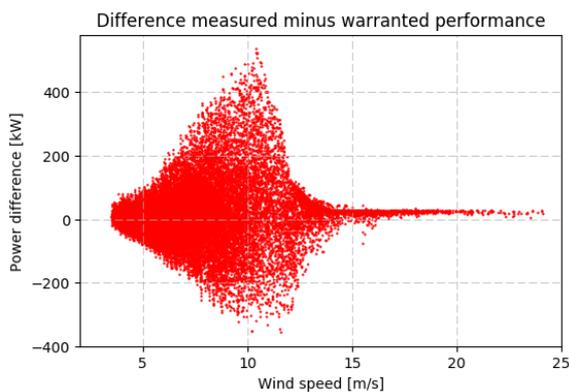
Hyper-parameter	Value
Architecture	[9 - 14 - 8 - 1]
Internal unknowns	341
Max. epochs	4000
Batch size	512
Test part of full dataset	25%
Layer activation functions	ELU
Output layer activation function	ELU/ Tanh
Optimization parameter	Adam
Learning rate	0.0002
Loss function	MAE
Early stopping epochs	50
Drop-out rate	None
Regularization L1 penalty	None

## Normalization scheme results

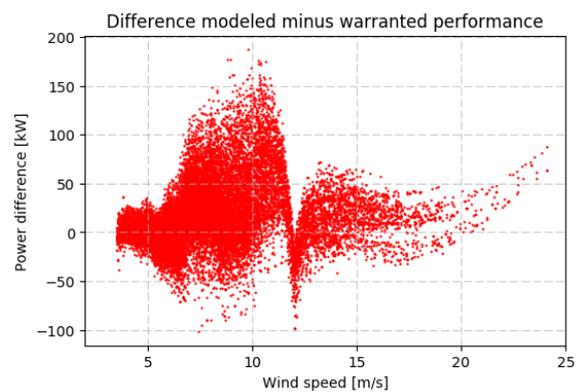
After training first the model needs to be validated. This shows whether the model is able to model after the input, the measured performance. If this is satisfactory, the performance of the models is quantified through the MAE, RMSE, and  $R^2$ -score. This allows for comparing normalization scheme performance to each other as well as to the warranted power curve method.

## Validation

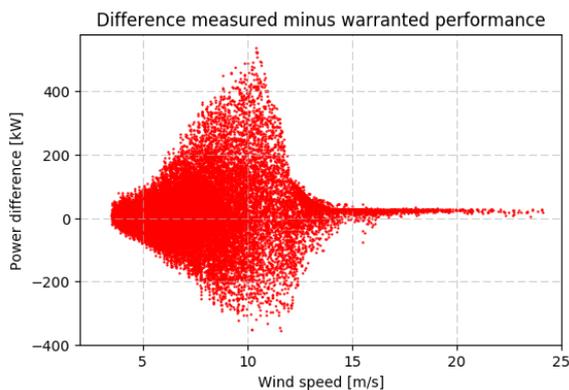
To validate the model the error/difference from the warranted power curve is plotted as is explained earlier. Then by analyzing the visible trends it can be seen whether or not the model is trained correctly and (globally) behaves as the measured performance as provided as input. Figure E-7 and Figure E-8 show the difference from the Warranted power curve (WPC) of the, respectively, measured performance and modeled performance for the normalization to the rated power scheme. The resulting difference from the WPC for the normalization to the deviation from the WPC is shown in Figure E-9 for the measured power and Figure E-10 for the modeled power. Figure E-7 and Figure E-9 are the same plot, but shown next to the modeled results for better comparison.



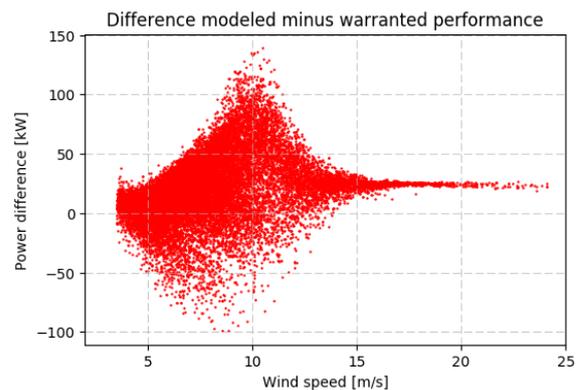
**Figure E-7:** Difference of measured power from warranted power for Prinses Alexia test set using a normalization to rated power scheme.



**Figure E-8:** Difference of modeled power from warranted power for Prinses Alexia test set using a normalization to rated power scheme.



**Figure E-9:** Difference of measured power from warranted power for Prinses Alexia test set using a normalization to difference from WPC scheme.



**Figure E-10:** Difference of modeled power from warranted power for Prinses Alexia test set using a normalization to difference from WPC scheme.

By comparing the left plot (from measured) to the right plot (from modeled) of normalization to rated power (Figure E-9 and E-10) something seems to go wrong near 12 m/s, near rated wind speed. Furthermore the right half of the plot is quite off from the measured performance at the left. Especially compared to the modeled results of the normalization to difference from WPC the performance of the model seems to be worse. Both have more or less the same range, but the trends in Figure E-8 are an indication this normalization scheme is not likely to result in an accurate model. To further substantiate this statement, the performance of the models themselves are quantified and compared next.

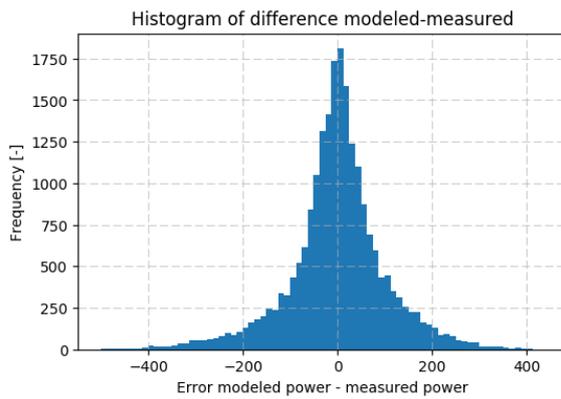
#### Performance evaluation

The performance of each model is quantified in the MAE and RMSE of error between measured and modeled performance and  $R^2$  score of measured to modeled power. This indicates how well the models relate to the measured data, provided as training data as was discussed earlier. The resulting statistical quantifications are shown and compared to the same statistics of the error between measured and warranted power in Table E-5. As can be seen in all aspects the normalization scheme of the difference with the warranted power performs best, followed by normalization to rated power and finally the warranted power curve method performs worst of these three.

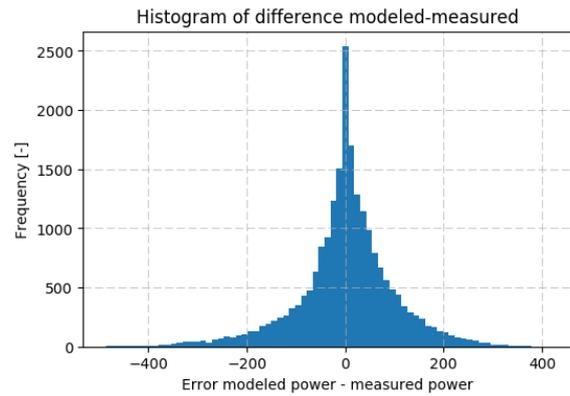
**Table E-5:** Resulting performance of trained model using only data from Prinses Alexia

	Normalized to rated power	Normalized to difference with warranted power	Warranted power
RMSE	102.66 kW	99.73 kW	105.46 kW
MAE	72.34 kW	69.29 kW	74.51 kW
$R^2$ -score	0.99118	0.99167	0.99069

To evaluate the performance of the model for each normalization scheme to the measurements, the difference between measured and modeled power, on which the statistical evaluation parameters in Table E-5 are based, is plotted in a histogram. This yields in more insight in the spread of the error, but also how symmetric this error is and where the highest peaks are located. The spread should of course be as low as possible and the peaks should be closest to 0. Figure E-11 shows this histogram for normalization to rated scheme, Figure E-12 the normalization to the difference from warranted power. As can be seen from comparing both histograms, the peak is for both schemes close to 0. However, the former normalization scheme is less slender and the (close to) 0 error peak is almost half of the normalization to difference from warranted power. This also shows the latter normalization scheme, being normalization to the difference from warranted power, performs better than the former scheme.



**Figure E-11:** Histogram of the difference between modeled and measured power for normalization to rated power, for the test dataset of Prinses Alexia.



**Figure E-12:** Histogram of the difference between modeled and measured power for normalization to the difference from warranted power, for the test dataset of Prinses Alexia.

### Concluding on normalization method

From the validation and the model performance evaluation it can be concluded the normalization to the difference from the warranted power curve is preferred to normalization to the rated power. The performance of the first is better when compared to the trends in the measurements as well as model performance comparison between these methods. Both, however, perform better than the method of only using the warranted power curve.

---

# Appendix F

---

## Grid search optimizations

In this Appendix grid searches for the Power curve verification (PCV) model and therefore also Performance Correlation Assessment (PCA) model are shown. From these grid searches the best hyper-parameters were selected. Just like with the 'generalizing' model the network lay-out is optimized via an extensive grid search, going over 800 cases. The result of this is shown in Section 6-4. From the regularization parameters, only the early stopping method is applied as a stopping criteria. The procedure is the same as described in Section 6-3.

The initial settings of the network, prior to starting the successive grid searches is shown in Table F-1. The grid searches are evaluated for the lowest Mean absolute error (MAE) and highest  $R^2$ -score for the test dataset (25% of the full set), as shown in Eq. (F-1) and Eq. (F-2) respectively. Due to dataset size, only Prinses Alexia is assessed for this model. The other wind farms have a dataset size which is considered too low to result consistently result in the global optimum, as no independence from stochastic behavior of the training function can be assumed.

**Table F-1:** Grid search hyper-parameters initial settings

Hyper-parameter	Value
Max. epochs	1000
Batch size	512
Test part of full dataset	25%
Hidden layer activation functions	Exponential linear unit (ELU)
Output layer activation function	Tanh
Optimization function	Adaptive moment estimation (Adam)
Learning rate	0.0002
Early stopping epochs	0.0 (no early stopping)
Drop-out rate	0.0 (no dropout)
Regularization penalty	0.0 (no penalty)
Training set	Prinses Alexia

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (\text{F-1})$$

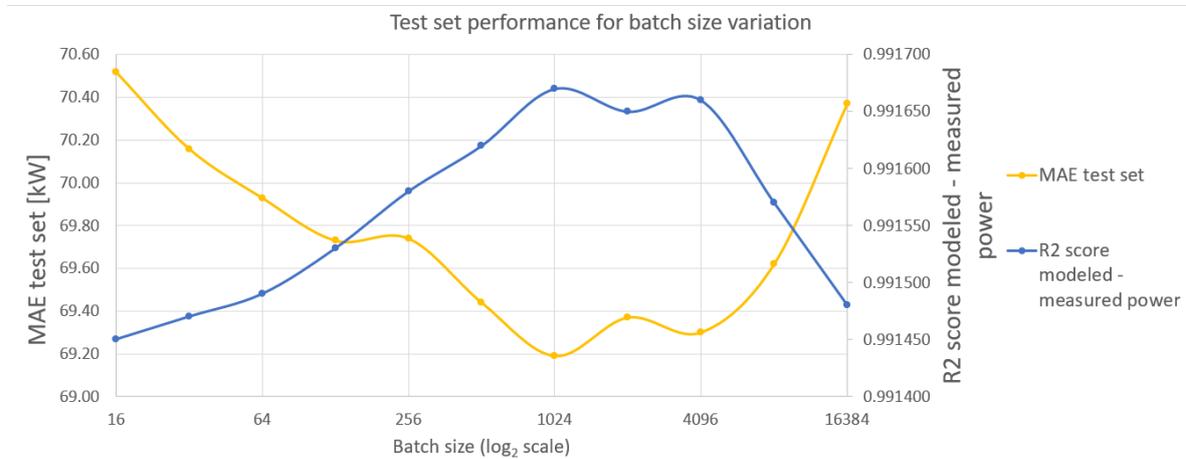
$$R^2 \equiv 1 - \frac{SS_{res}}{SS_{tot}} \quad (\text{F-2})$$

$$SS_{res} = \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (\text{F-3})$$

$$SS_{tot} = \sum_{i=1}^N \left( y_i - \frac{1}{N} \sum_{i=1}^N y_i \right)^2 \quad (\text{F-4})$$

## F-1 Epoch and batch size

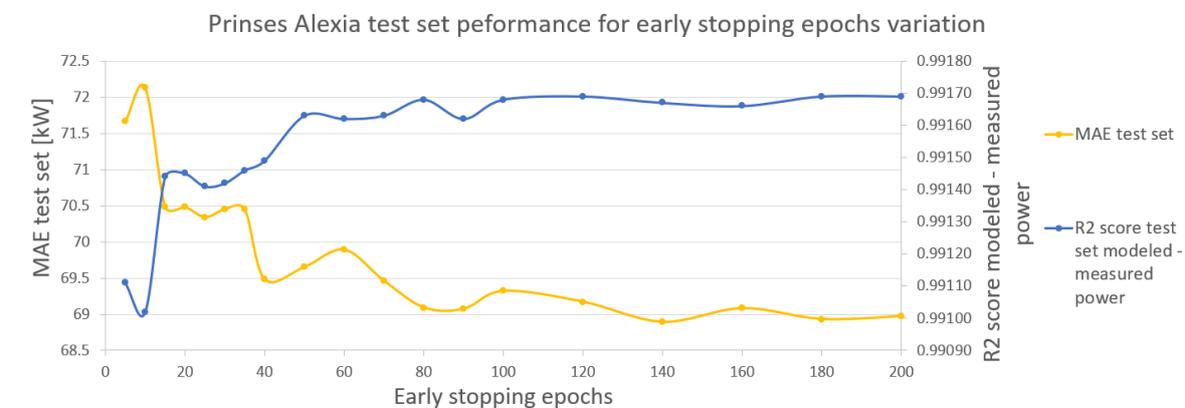
First the batch size is evaluated in the same manner as done in Section 6-3, with the same variation in batch size being powers of 2 ranging from 16 to 16384. The resulting variation of both the MAE and  $R^2$ -score is shown in Figure F-1. As can be seen there is a clear optimal value in both the MAE and  $R^2$ -score, which both correspond to a batch size of 1024 samples. Therefore this is set as batch size and the next hyper-parameter is optimized, the early stopping epoch number.



**Figure F-1:** PCV model batch size grid search, evaluated for the test dataset of Prinses Alexia.

## F-2 Early stopping epochs

The final hyper-parameter to be determined for the PCV model is the early stopping epoch number. To evaluate this the maximum amount of epochs is set sufficiently large in order to never reach this maximum. This results in the early stopping criteria to be the only stopping criteria. The early stopping epochs range from 5 to 200 in steps of 5 epochs up to 40 early stopping epochs, after which this increases to 10 epochs. The change in early stopping epochs is the same as described in Section 6-3. Figure F-2 shows the resulting variation of the MAE and  $R^2$ -score for the variation of the early stopping epochs. The model seems to enter a converged state after about 120/140 epochs. To be on the safe side of this estimation, the early stopping amount is set at 140 epochs.



**Figure F-2:** Test dataset performance for varying early stopping epochs for Prinses Alexia.



# Numerical sensitivity and correlation on measurements

In Section 7-2 the concept of Numerical sensitivity analysis (NSA) was introduced and applied to the dataset of Prinses Alexia, to determine the sensitivity of individual parameter along the turbine power curve. Comparable to this sensitivity analysis is the more known Pearson correlation coefficient. Recall this coefficient was not of primary interest since the Pearson correlation assumed a normal, not necessarily zero-mean, distribution which was not necessarily the case for the measured data. In this Appendix both methods are compared, followed by the Pearson correlation coefficient of all binned parameters.

## G-1 Correlation coefficient

The Pearson correlation coefficient,  $\rho$ , is calculated using Eq. (G-1). The correlation coefficient ranges from  $-1$  to  $1$ , where  $1$  indicates a 'perfect' increasing linear relationship, while  $-1$  indicates a 'perfect' decreasing linear relation. [69]

$$\rho_{X,Y} = \text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \quad (\text{G-1})$$

Where:

- $E$  represents the expected value, as defined in equation (G-2) [70]
- $\text{cov}$  represents the covariance, defined as  $E[(X - \mu_X)(Y - \mu_Y)]$
- $\sigma$  represents standard deviation

$$E[X] = x_1 p_1 + x_2 p_2 + \dots + x_n p_n \quad (\text{G-2})$$

Where:

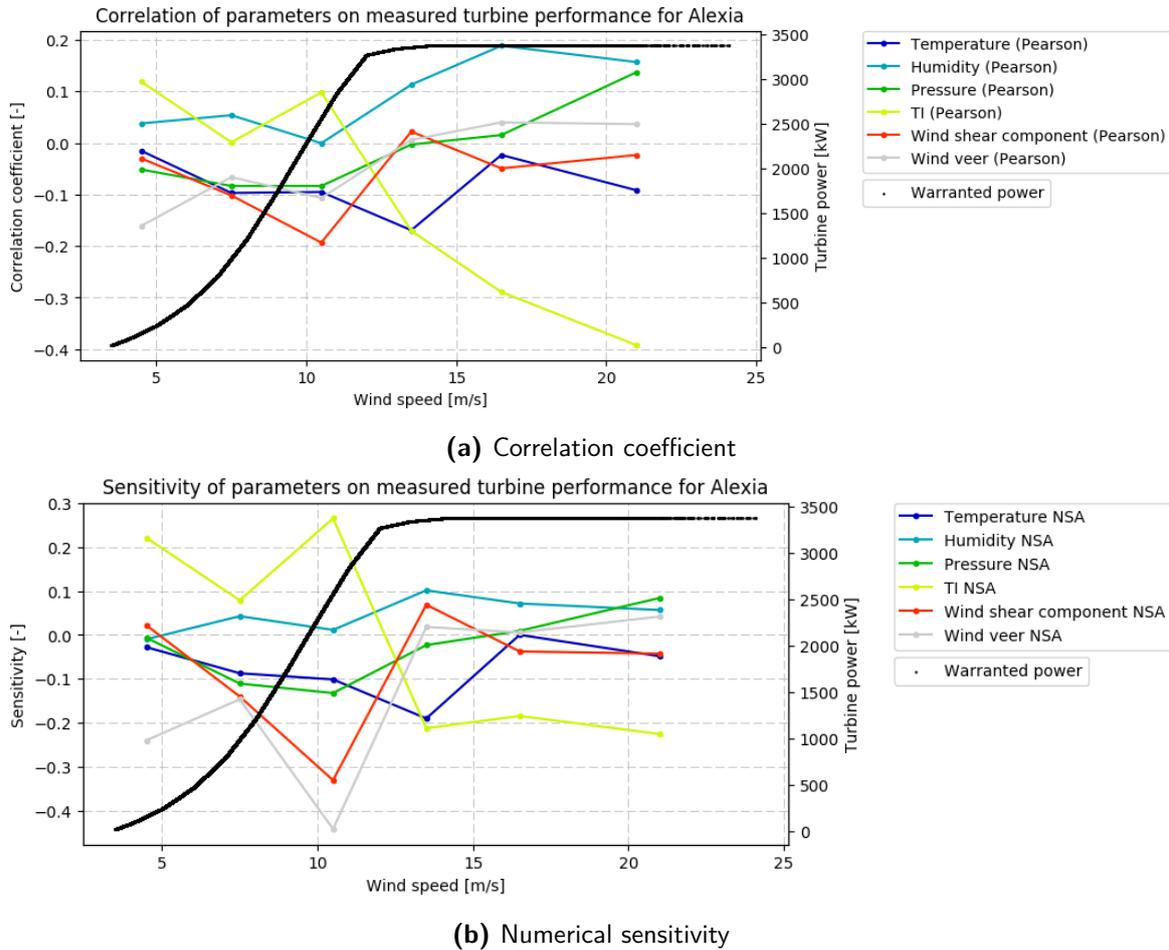
- X represents a parameter with  $x_1, x_2 \dots x_n$  outcomes/values.
- $p_1, p_2 \dots p_n$  represent the probabilities of the outcomes/values in X.

## G-2 Numerical sensitivity versus correlation comparison

While the numerical sensitivity and correlation coefficient indicate about the same relation, the method is different. This might result in different values, and therefore insight, in relations between an atmospheric parameter and turbine performance. In this section both methods are compared, first on individual parameters binned for only velocity and to be plotted along the power curve, then also binned for several magnitudes of a single variable, as done in Section 7-2. Note that the analyses below are only performed for measured data, no modeling is introduced. The limitations of this were already mentioned in Section 7-2. Due to the required size the analyses are done only for Prinses Alexia. The dataset size also limits the velocity bin width, which is set at 3 m/s to have (mostly) sufficient samples per bin. However, for higher wind speeds the amount of samples is considerably less, rendering these calculated correlations less reliable.

### Split velocity bins

Figure G-1 shows the propagation of both correlation coefficient (a) and numerical sensitivity (b) of individual parameters only split into velocity bins. In general both correlations are fairly coinciding, except some outlying numerical sensitivities for Turbulence intensity (TI), wind shear and wind veer near 10 m/s. Note that in this analysis no additional filtering is performed to single out individual parameters. The full dataset is only split in wind velocity bins, which are then assessed for numerical sensitivity and correlation. Additionally influence of the magnitude of investigated parameters can be studied, which is performed next.



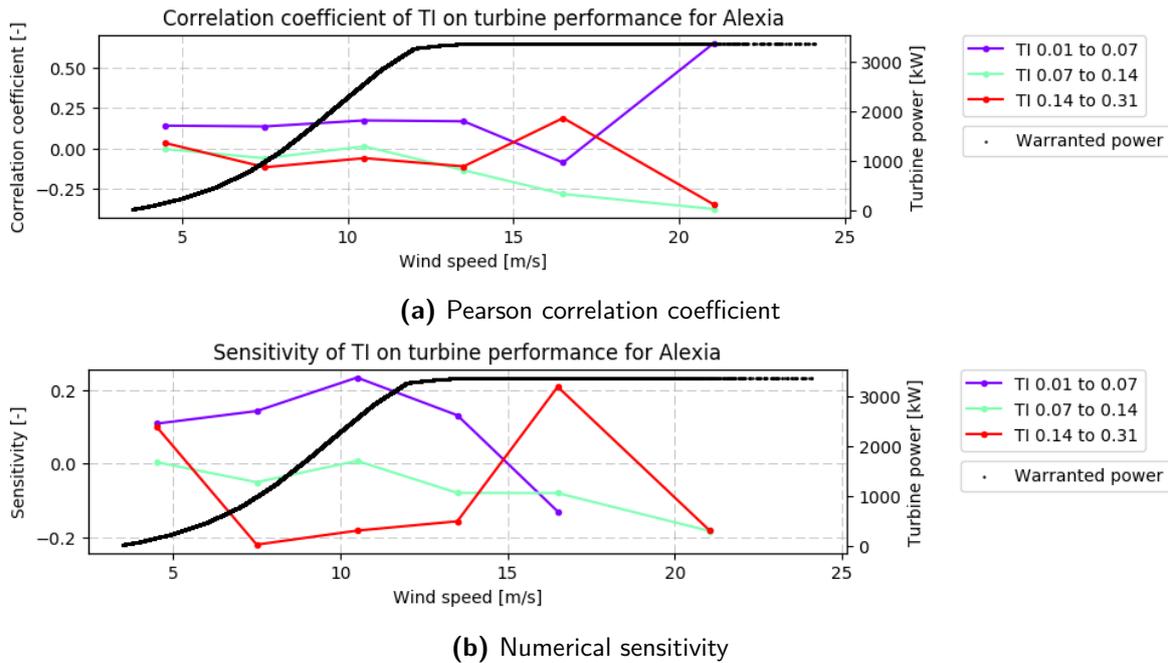
**Figure G-1:** Pearson correlation (a) and numerical sensitivity (b) of individual parameters over the course of the power curve, right axis, for Prinses Alexia.

### Split parameter magnitude

Splitting not only for wind speed but also for the magnitude of the investigated parameter additionally allows for studying the effect of the parameter magnitude on the correlation to the turbine performance. This is done by splitting the dataset into three about evenly sized subsets, each having a range of a single parameter which is studied. Subsequently the numerical sensitivity and correlation coefficient are calculated for each range at each wind speed bin. Besides studying the effect of a high or low value of this parameter, for example, it allows for a better comparison of the correlation coefficient and NSA. This is done below for all parameters excluding wind speed. Note that no additional filtering is done on the dataset and all other variables are not constant, but are still varying.

**Turbulence intensity** Figure G-2 shows both the correlation coefficient as the numerical sensitivity of the TI. Both methods seem to result in approximately the same correlation (note the y-axis scale difference), except near the end. This difference in the end is likely caused due to a lack of sufficient data, rendering the correlation less accurate and reliable as

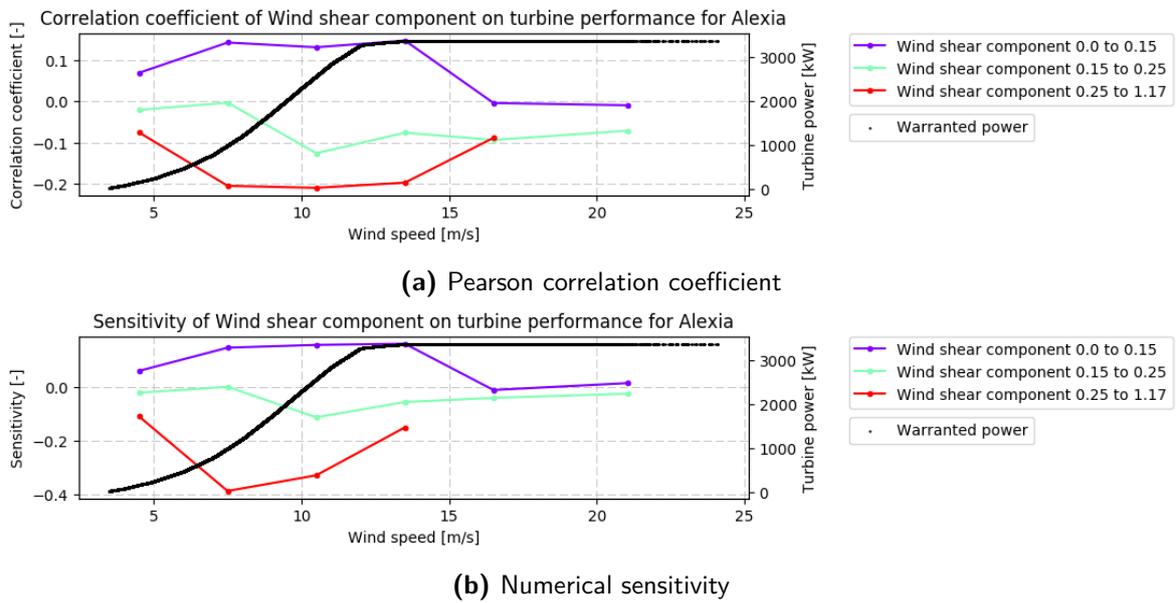
mentioned earlier. A high correlation is not expected as the performance barely changes. Low turbulence tends to have a prolonged positive correlation with turbine performance, which is consistent with literature, while high turbulence starts positive, but quickly changes to a stronger negative correlation. Medium TI, indicated by the cyan line, seems to be about in the middle of both low and high TI, having no distinct correlation/sensitivity to turbine performance or having positive and negative influences cancel out. Towards high wind speed, the impact is negative for all magnitude ranges of TI, again in line with literature.



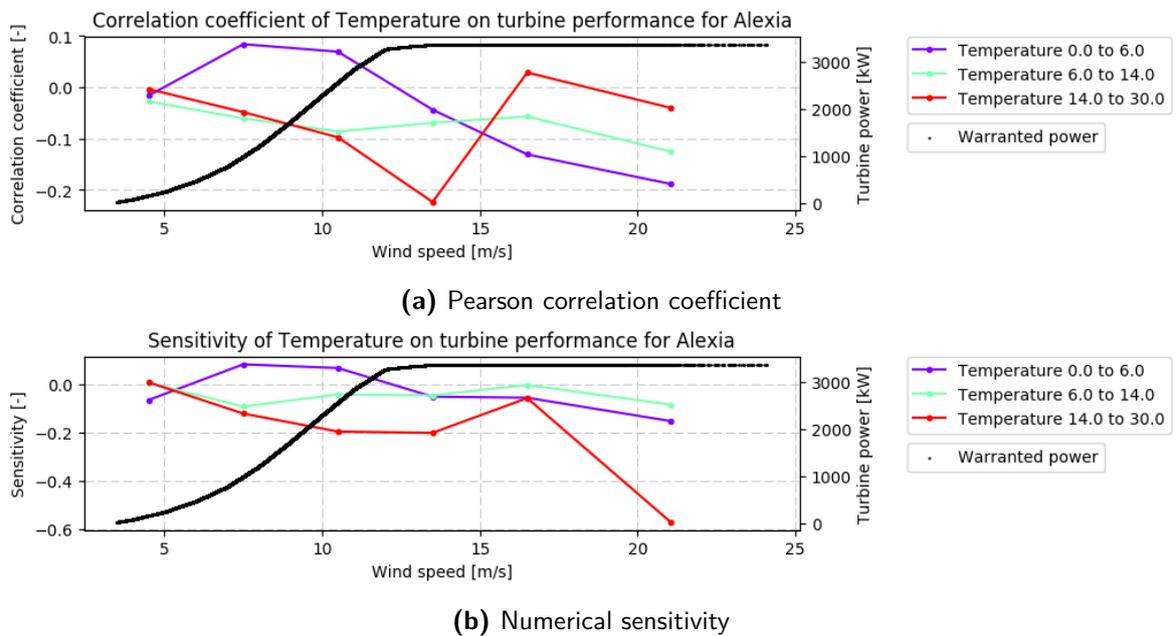
**Figure G-2:** Correlation and numerical sensitivity of turbulence intensity along power curve for Prinses Alexia measured data.

**Wind shear** The correlation coefficient and numerical sensitivity for the wind shear, as shown in Figure G-3, also correspond with each other, with the exception of the magnitude of the numerical sensitivity of high wind shear. Additionally low wind shear has a clear positive correlation with turbine performance, while high wind shear has a strong negative correlation. In between the correlation is slightly negative, though close to 0.

**Temperature** Figure G-4 shows the comparison of the correlation coefficient and numerical sensitivity for the ambient temperature. Both methods are very similar with the exception, again, for high wind speed. The general correlation is that temperature has a negative influence on turbine performance, a negative correlation. This also corresponds with the theory, as temperature negatively affects density, which in turn affects wind power flux positively.

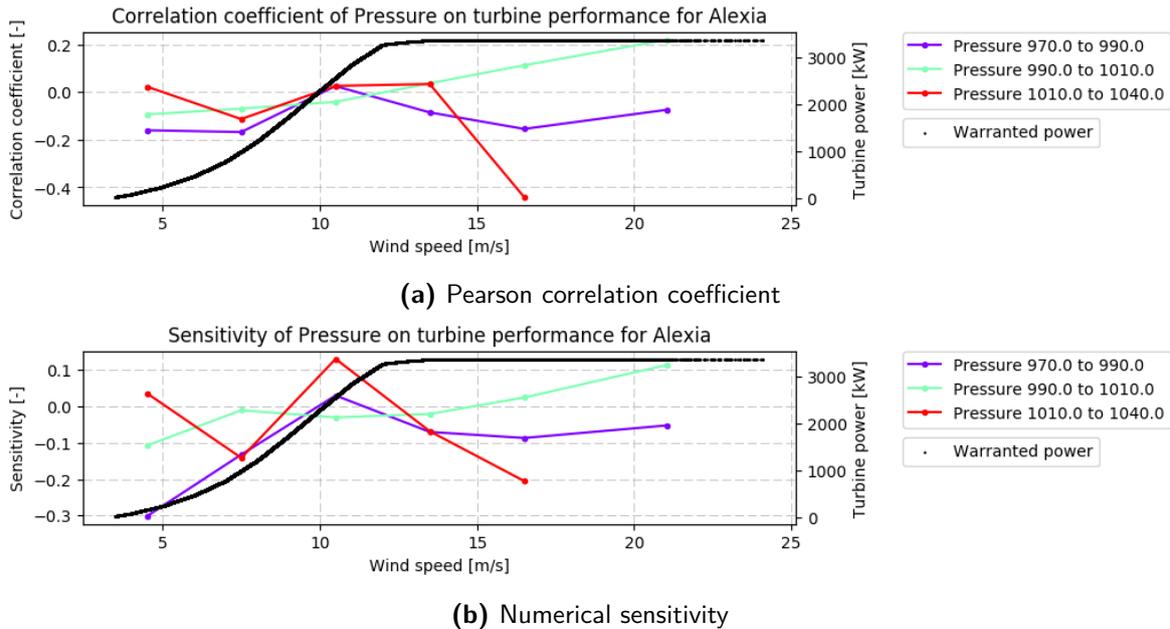


**Figure G-3:** Correlation and numerical sensitivity of wind shear along power curve for Prinses Alexia measured data.



**Figure G-4:** Correlation and numerical sensitivity of ambient temperature along power curve for Prinses Alexia measured data.

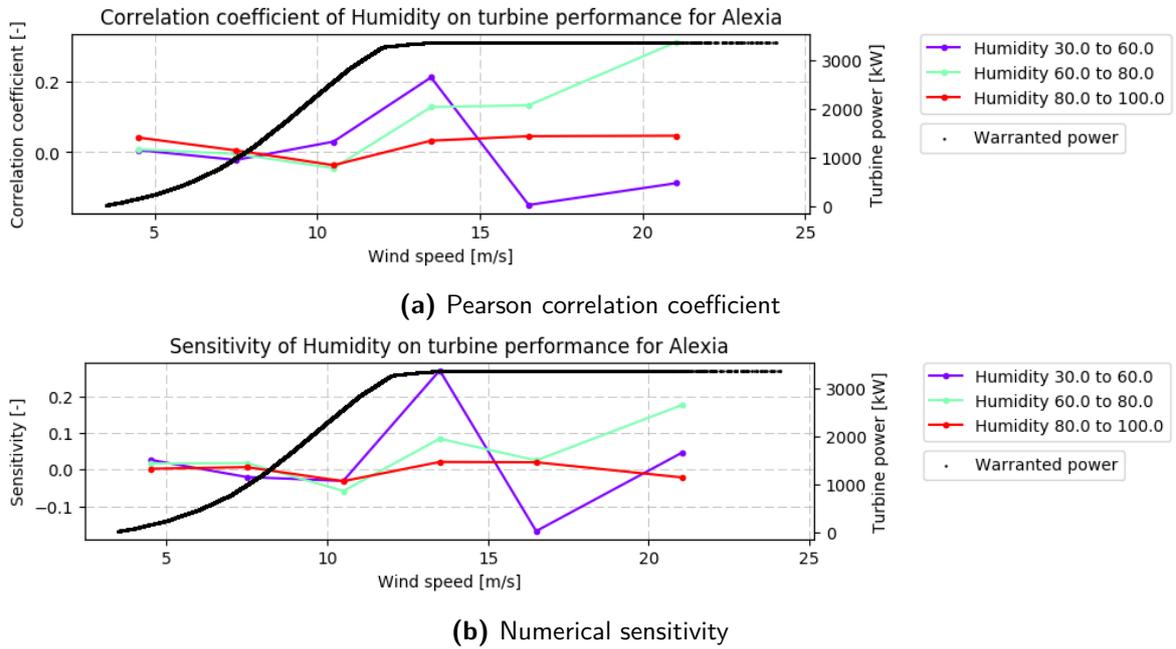
**Pressure** Comparing the numerical sensitivity and correlation coefficient for ambient pressure on turbine performance, shown in Figure G-5, both methods coincide quite well. As the correlation of all three pressure ranges is quite close, it can be assumed there is barely any influence of the magnitude of ambient pressure on the influence on turbine performance. This influence also tends to be small as it almost oscillates close to 0, with again an exception for high wind speed. For high pressure (red line) no subset exists for high wind speed.



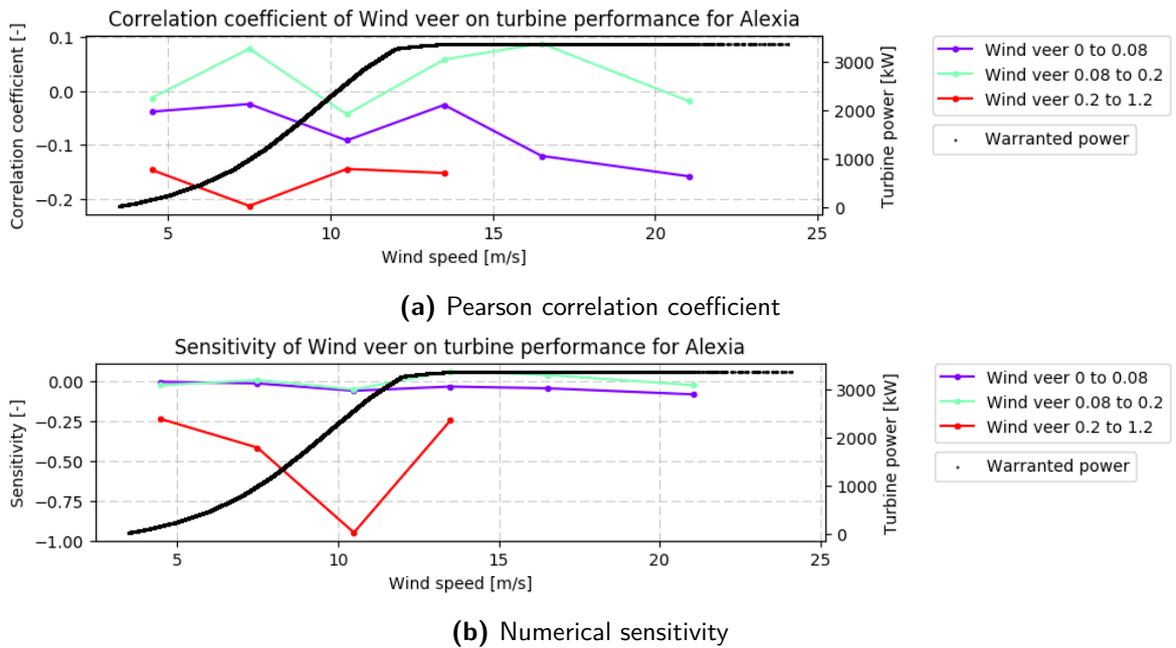
**Figure G-5:** Correlation and numerical sensitivity of turbulence intensity along power curve for Prinses Alexia measured data.

**Humidity** The correlation coefficient and numerical sensitivity of relative humidity is shown in Figure G-6. As can be seen both methods follow -more or less- the same trend, starting with almost no correlation, increasing to positive correlation near rated wind speed. However, as the low humidity bin is small (the average measured humidity is 80%) the found correlations might be off, which could explain the sudden oscillation for low humidity at higher wind speeds.

**Wind veer** Figure G-7 shows the correlation coefficient and numerical sensitivity of the wind veer to turbine performance. The spread of wind veer in the dataset was very unilateral, as the large majority was close to  $0^\circ/m$ . This can be seen in the short red line for relative high wind veer. This also limits the accuracy and reliability of medium wind veer, the cyan line, which tends to oscillate around 0 correlation. Low wind veer, the blue line, being most accurate and reliable shows a slight negative correlation to turbine performance. Wind veer results in a less than ideal inflow angle over the rotor, which should result in a negative influence on performance. This can be seen from the blue line, with a slightly negative correlation.



**Figure G-6:** Correlation and numerical sensitivity of relative humidity along power curve for Prinses Alexia measured data.



**Figure G-7:** Correlation and numerical sensitivity of wind veer along power curve for Prinses Alexia measured data.



---

## Appendix H

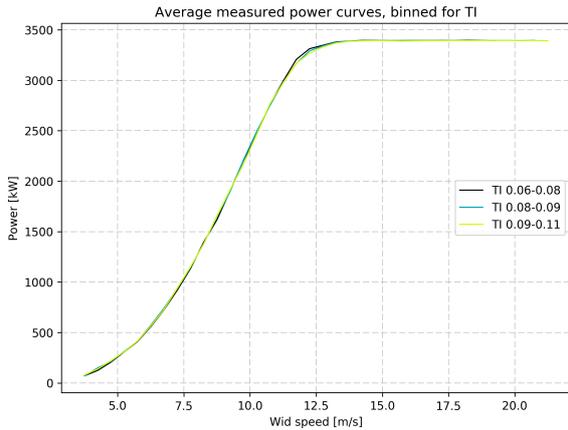
---

# Singled out parameter impact on power curve

Validation of the Performance Correlation Assessment (PCA) model was performed in Section 7-1 and 7-2, where the former validated the performance of the model settings and the latter validated visible trends with literature, as far as possible. For further (although crude) validation of this model, power curves for certain ranges of a single parameter are plotted. This means that the full dataset is split for certain ranges of a parameter under investigation and the average performance shift is plotted, resulting in 3 or 4 power curves each with a different range of a single parameter. For the modeled results all other parameters, with the exception of the wind speed, are kept constant (as explained in Section 7-2), however, the dataset size of Prinses Alexia does not allow this as the filtered dataset becomes too small. For this reason the range of all other variables, otherwise set constant, is to be taken  $\pm\sigma$  from the mean value in order to construct three power curves for the mentioned ranges of the parameter to be investigated. Despite this wide range, some power curves might miss some sections as can be seen below. From each dataset, the power curve is constructed using the Method of Bins (MoB). Despite other variables are not constant by far, some trends in splitting the range of a single parameter might be identified and be found also in the power binned power curves from the model. It does, however, limit the validity of the power curves from the measured dataset. Each set of power curves is discussed per parameter below.

### Turbulence intensity

The binned power curves for Turbulence intensity (TI) from the Prinses Alexia measured dataset is shown in Figure H-1, the results from the trained model using the generic input dataset on Figure H-2. Barely any trend can be identified from the measured power curves, except a slight negative correlation at the knee of the power curve(s), which is in line with literature. The power curves from the model also show this trend, but more stronger present. They also show a positive impact for low wind velocities, up to about 10 m/s.



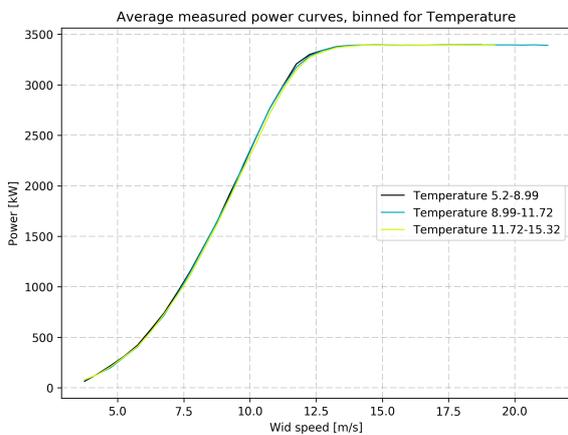
**Figure H-1:** Individual influence of TI on performance for Prinses Alexia via measurements.



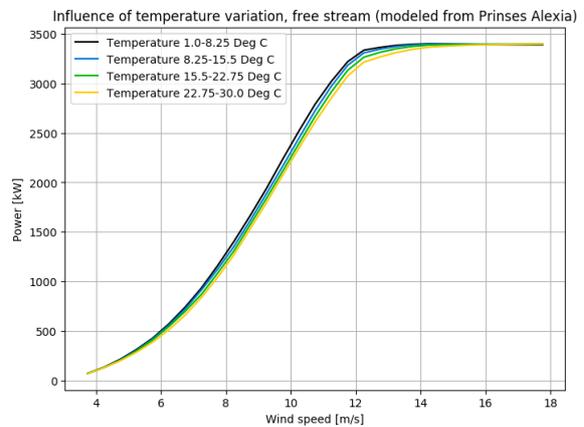
**Figure H-2:** Individual influence of TI on performance for Prinses Alexia via model.

### Ambient temperature

There is barely any trend visible from the measured power curves binned for temperature (Figure H-3), except a small negative correlation near rated power. More clearly visible in the power curves from the model (Figure H-4) is the negative impact of higher temperature over the whole wind speed range. This can be explained through lower density, caused by higher temperature, which lowers the wind power flux linearly.



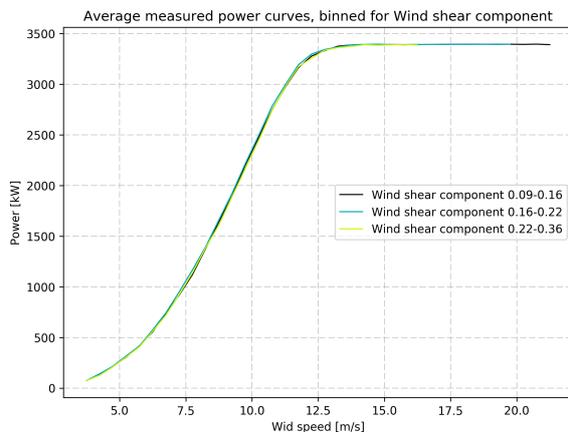
**Figure H-3:** Individual influence of temperature on performance for Prinses Alexia via measurements.



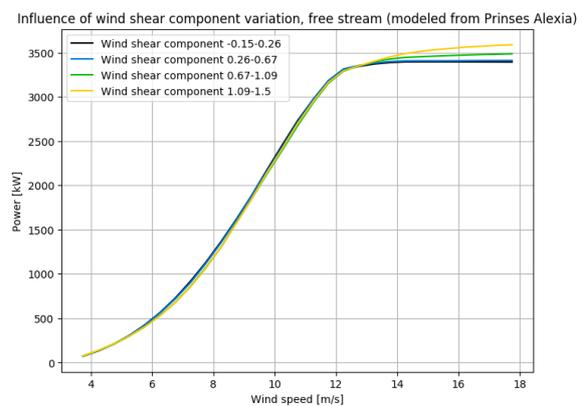
**Figure H-4:** Individual influence of temperature on performance for Prinses Alexia via model.

## Wind shear

From the binned power curves for wind shear constructed out of the measured data, shown in Figure H-5, a very slight negative impact can be seen. The same can be seen for the results from the model, in Figure H-6, although the difference is too small for both to make any statement with certainty. What seems to be out of line is the strong positive impact stronger wind shear has for the constant power part of the power curve. The likely cause of this is a lack of sufficient training data for this region (high wind speed and medium to high wind shear). This is also seen in Figure G-4, in Appendix G, as no data for high wind shear exists for high wind speed.



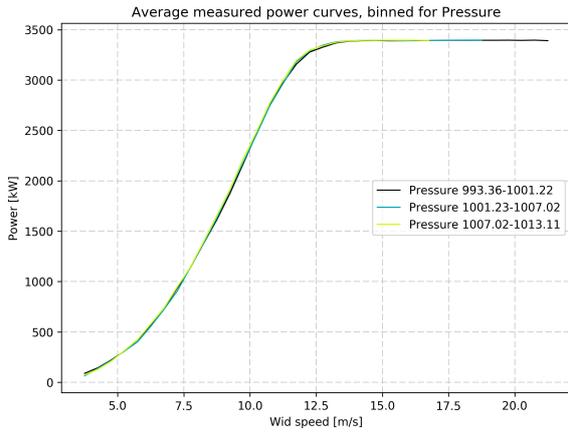
**Figure H-5:** Individual influence of wind shear on performance for Prinses Alexia via measurements.



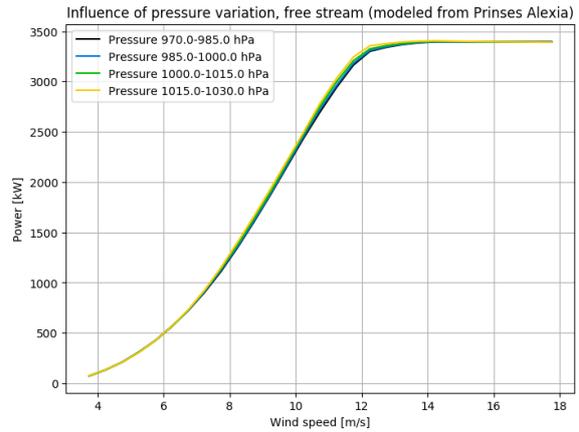
**Figure H-6:** Individual influence of wind shear on performance for Prinses Alexia via model.

## Ambient pressure

The measured binned power curves for pressure shown in Figure H-7, are too close to each other to identify a global trend of the influence of pressure on the turbine performance. The modeled power curves in Figure H-8 do show a slight trend as an increase in pressure results in higher turbine performance. This can be explained because of an increase in pressure (linearly) increases the density, which in turn increases the wind power flux.



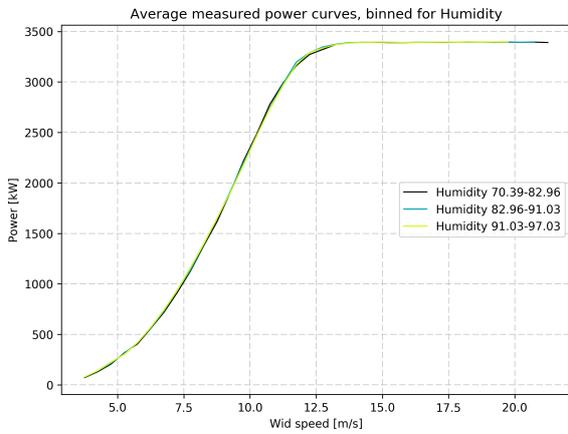
**Figure H-7:** Individual influence of pressure on performance for Prinses Alexia via measurements.



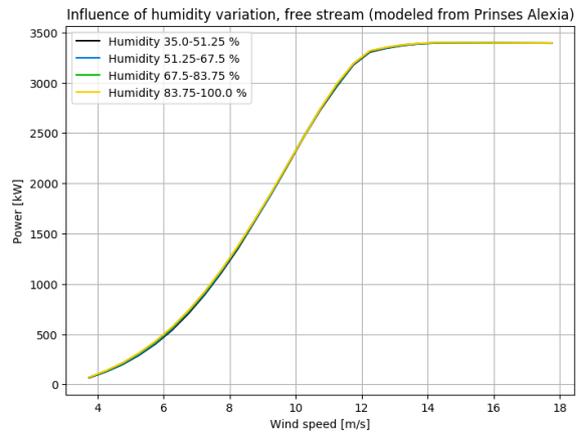
**Figure H-8:** Individual influence of pressure on performance for Prinses Alexia via model.

## Humidity

The binned power curves for humidity for measured and modeled turbine performance, respectively Figure H-9 and Figure H-10, show no clear trends to be identified. Although the power curves for the model shift slightly, no statement can be given with certainty.



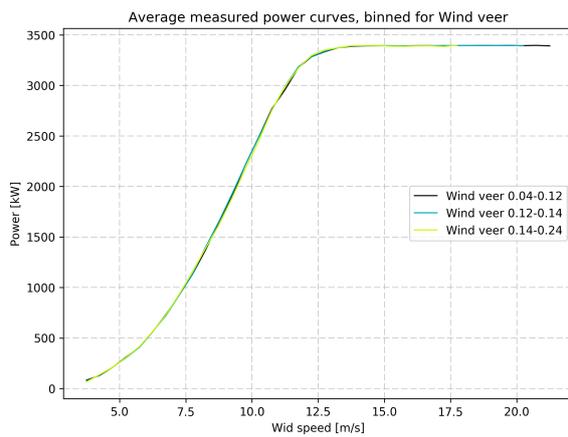
**Figure H-9:** Individual influence of humidity on performance for Prinses Alexia via measurements.



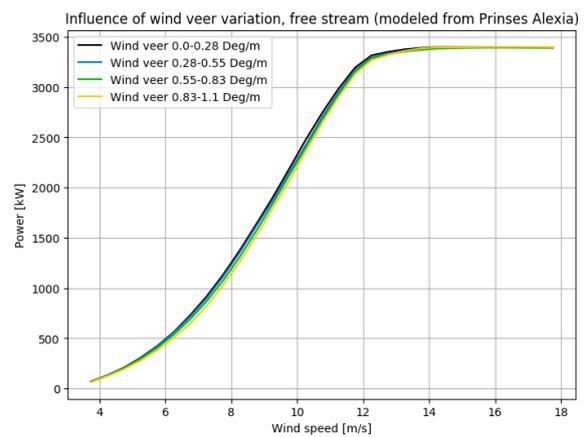
**Figure H-10:** Individual influence of relative humidity on turbine performance for Prinses Alexia

## Wind veer

Finally, in the power curves for variations in the wind veer (Figure H-11 for measured dataset, Figure H-12 for modeled power) a slight decrease in turbine performance for increased wind veer can be identified, which is in line with theory. The effect is larger, however, for the modeled power which is also caused by the larger ranges of the power curve bins. As there were very few data samples in the measured dataset for relative high wind veer variations, the range of these power curves -especially the maximum range- is significantly smaller than those from the model.



**Figure H-11:** Individual influence of wind veer on performance for Prinses Alexia via measurements.



**Figure H-12:** Individual influence of wind veer on performance for Prinses Alexia via model.



---

# Bibliography

- [1] F. Pelletier, C. Masson, and A. Tahan, “Wind turbine power curve modelling using artificial neural network,” Renewable Energy, vol. 89, pp. 207–214, 2016.
- [2] O. Janssens, N. Noppe, C. Devriendt, R. Van de Walle, and S. Van Hoecke, “Data-driven multivariate power curve modeling of offshore wind turbines,” Engineering Applications of Artificial Intelligence, vol. 55, pp. 331–338, 2016.
- [3] G. W. E. Council, “Global wind report - annual market update 2016,” report, Global Wind Energy Council, Rue d’Arlon 80 1040 Brussels, Belgium, 2016.
- [4] G. Lee, Y. Ding, M. G. Genton, and L. Xie, “Power curve estimation with multivariate environmental factors for inland and offshore wind farms,” Journal of the American Statistical Association, vol. 110, no. 509, pp. 56–67, 2015.
- [5] PCWG, “Power curve working group website.” Website, 2017.
- [6] K. Kaiser, W. Langreder, H. Hohlen, and J. Højstrup, “Turbulence correction for power curves,” Wind Energy, pp. 159–162, 2007.
- [7] K. Walter, C. C. Weiss, A. H. Swift, J. Chapman, and N. D. Kelley, “Speed and direction shear in the stable nocturnal boundary layer,” Journal of Solar Energy Engineering, vol. 131, no. 1, p. 011013, 2009.
- [8] C. Tsalicoglou, S. Barber, N. Chokani, and R. S. Abhari, “Effect of flow inclination on wind turbine performance,” Journal of Engineering for Gas Turbines and Power, vol. 134, no. 12, p. 122601, 2012.
- [9] H. Wu, Y. Zhou, Q. Luo, and M. A. Basset, “Training feedforward neural networks using symbiotic organisms search algorithm,” Computational Intelligence and Neuroscience, vol. 2016, 2016.
- [10] S. Ruder, “An overview of gradient descent optimization algorithms,” arXiv preprint arXiv:1609.04747, 2016.

- [11] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in International conference on machine learning, pp. 1139–1147, 2013.
- [12] D. Mishkin, N. Sergievskiy, and J. Matas, "Systematic evaluation of cnn advances on the imagenet," arXiv preprint arXiv:1606.02228, 2016.
- [13] S. Mc Loone and G. Irwin, "Improving neural network training solutions using regularization," Neurocomputing, vol. 37, no. 1, pp. 71–90, 2001.
- [14] C. IEC, "61400-12-1: Wind turbines - part 12-1: Power performance measurements of electricity producing wind turbines," Geneva, Switzerland: IEC, 2017. -CONFIDENTIAL-.
- [15] F. A. Luce, "Test report wico 247lkc14/01," power curve test report, WIND-consult GmbH, Reuterstrase 9 D-18211 Bargeshagen, 04 2015. -CONFIDENTIAL-.
- [16] M. Andreas, "Power performance test report - pyc windfarm," techreport GLGH 4270 15 13064 290-A-0006, DNV GL, 09 2017. DRAFT/ -CONFIDENTIAL-.
- [17] G. Demurtas, "Met-mast in norrekaer enge wind farm," 10 2014. -CONFIDENTIAL-.
- [18] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," Statistics and computing, vol. 14, no. 3, pp. 199–222, 2004.
- [19] J.-S. Jang, "Anfis: adaptive-network-based fuzzy inference system," IEEE transactions on systems, man, and cybernetics, vol. 23, no. 3, pp. 665–685, 1993.
- [20] M. Schlechtingen, I. F. Santos, and S. Achiche, "Using data-mining approaches for wind turbine power curve monitoring: a comparative study," IEEE Transactions on Sustainable Energy, vol. 4, no. 3, pp. 671–679, 2013.
- [21] G. N. Thorben, "Pen y cymoedd power curve warranty for the swt 3.0-08," warranty contract, DNVGL, 01 2017. -CONFIDENTIAL-.
- [22] "Wind farm production assessment for norrekaer enge wind farm (denmark)," Tech. Rep. 0, Barlovento Recursos Naturales, Pintor Sorolla 8, 1.ªA. 26007 LOGROÑO (SPAIN), 03 2015. -CONFIDENTIAL-.
- [23] M. Andreas, "As-built report for site calibration, pen y cymoedd wind farm [turbine specific]," report, DNV-GL, 05 2016. -CONFIDENTIAL-.
- [24] C. J. Marchena, "Assessment of the validity of the warranted power curve in nonstandard wind conditions," msc. thesis, Technical University Delft, June 2017.
- [25] S. Shokrzadeh, M. J. Jozani, and E. Bibeau, "Wind turbine power curve modeling using advanced parametric and nonparametric methods," IEEE Transactions on Sustainable Energy, vol. 5, no. 4, pp. 1262–1269, 2014.
- [26] A. Abraham, "Adaptation of fuzzy inference system using neural learning," Fuzzy systems engineering, pp. 914–914, 2005.

- 
- [27] F. Chollet et al., “Keras.” <https://github.com/fchollet/keras>, 2015.
- [28] A. Kirby, “Energy: Meeting soaring demand.” Internet, 11 2004.
- [29] A. Kusiak and H. Zheng, “Optimization of wind turbine energy and power factor with an evolutionary computation algorithm,” Energy, vol. 35, no. 3, pp. 1324–1332, 2010.
- [30] A. Kusiak, H. Zheng, and Z. Song, “Power optimization of wind turbines with data mining and evolutionary computation,” Renewable energy, vol. 35, no. 3, pp. 695–702, 2010.
- [31] A. Albers and C. Hinsch, “Influence of different meteorological conditions on the power performance of large wecs,” DEWI Magazin, vol. 9, pp. 40–49, 1996.
- [32] A. Albers, T. Jakobi, R. Rohden, and J. Stoltenjohannes, “Influence of meteorological variables on measured wind turbine power curves,” in The European Wind Energy Conf. & Exhibition, pp. 525–546, 2007.
- [33] S. Wharton and J. K. Lundquist, “Atmospheric stability affects wind turbine power collection,” Environmental Research Letters, vol. 7, no. 1, p. 014005, 2012.
- [34] D. van der Arend, “Thesis literature study.” 07 2017.
- [35] T. Burton, N. Jenkins, D. Sharpe, and E. Bossanyi, Wind energy handbook. John Wiley & Sons, 2011.
- [36] F. A. Jowder, “Wind power analysis and site matching of wind turbine generators in kingdom of bahrain,” Applied Energy, vol. 86, no. 4, pp. 538–545, 2009.
- [37] PCWG, “Pcwg 1st sharing initiative definition.” Online, 2015.
- [38] J. F. Manwell, J. G. McGowan, and A. L. Rogers, Wind energy explained: theory, design and application. John Wiley & Sons, 2010.
- [39] I. Antoniou, S. M. Pedersen, and P. B. Enevoldsen, “Wind shear and uncertainties in power curve measurement and wind resources,” Wind Engineering, vol. 33, no. 5, pp. 449–468, 2009.
- [40] E. Rareshide, A. Tindal, C. Johnson, A. Graves, E. Simpson, J. Bleeg, T. Harris, and D. Schoborg, “Effects of complex wind regimes on turbine performance,” in Proceedings American Wind Energy Association WINDPOWER Conference (Chicago, IL), 2009.
- [41] D. L. Elliott and J. B. Cadogan, “Effects of wind shear and turbulence on wind turbine power curves,” tech. rep., Pacific Northwest Lab., Richland, WA (USA), 1990.
- [42] J. Sumner and C. Masson, “Influence of atmospheric stability on wind turbine power performance curves,” Journal of solar energy engineering, vol. 128, no. 4, pp. 531–538, 2006.
- [43] K. S. Hansen, R. J. Barthelmie, L. E. Jensen, and A. Sommer, “The impact of turbulence intensity and atmospheric stability on power deficits due to wind turbine wakes at horns rev wind farm,” Wind Energy, vol. 15, no. 1, pp. 183–196, 2012.

- [44] M. Dörenkämper, J. Tambke, G. Steinfeld, D. Heinemann, and M. Kühn, “Atmospheric impacts on power curves of multi-megawatt offshore wind turbines,” in Journal of Physics: Conference Series, vol. 555, p. 012029, IOP Publishing, 2014.
- [45] W. Brutsaert, Evaporation into the atmosphere: theory, history and applications, vol. 1. Springer Science & Business Media, 2013.
- [46] W. Yue, Y. Xue, and Y. Liu, “High humidity aerodynamic effects study on offshore wind turbine airfoil/blade performance through cfd analysis,” International Journal of Rotating Machinery, vol. 2017, 2017.
- [47] M. Lydia, A. I. Selvakumar, S. S. Kumar, and G. E. P. Kumar, “Advanced algorithms for wind turbine power curve modeling,” IEEE Transactions on sustainable energy, vol. 4, no. 3, pp. 827–835, 2013.
- [48] M. Lydia, S. S. Kumar, A. I. Selvakumar, and G. E. P. Kumar, “A comprehensive review on wind turbine power curve modeling techniques,” Renewable and Sustainable Energy Reviews, vol. 30, pp. 452–460, 2014.
- [49] S. Li, D. C. Wunsch, E. A. O’Hair, and M. G. Giesselmann, “Using neural networks to estimate wind turbine power generation,” IEEE Transactions on energy conversion, vol. 16, no. 3, pp. 276–282, 2001.
- [50] R. Veena, V. Femin, S. Mathew, I. Petra, and J. Hazra, “Intelligent models for the power curves of small wind turbines,” in Cogeneration, Small Power Plants and District Energy (ICUE), International Conference on, pp. 1–5, IEEE, 2016.
- [51] A. Kusiak, H. Zheng, and Z. Song, “Models for monitoring wind farm power,” Renewable Energy, vol. 34, no. 3, pp. 583–590, 2009.
- [52] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in Proceedings of the IEEE international conference on computer vision, pp. 1026–1034, 2015.
- [53] B. T. Polyak, “Some methods of speeding up the convergence of iteration methods,” USSR Computational Mathematics and Mathematical Physics, vol. 4, no. 5, pp. 1–17, 1964.
- [54] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.
- [55] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, “On large-batch training for deep learning: Generalization gap and sharp minima,” arXiv preprint arXiv:1609.04836, 2016.
- [56] M. A. Nielsen, Neural networks and deep learning. Determination Press USA, 2015.
- [57] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting.,” Journal of machine learning research, vol. 15, no. 1, pp. 1929–1958, 2014.

- 
- [58] A. M. Society, “Meteorology glossary.” Internet, 2012. <http://glossary.ametsoc.org/wiki/Supersaturation>.
- [59] G. van Rossum, “Python tutorial,” technical report cs-r9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, May 1995.
- [60] R Core Team, R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, 2013.
- [61] N. R. Draper and H. Smith, Applied regression analysis. John Wiley & Sons, 2014.
- [62] J. Montano and A. Palmer, “Numeric sensitivity analysis applied to feedforward neural networks,” Neural Computing & Applications, vol. 12, no. 2, pp. 119–125, 2003.
- [63] V. Vapnik, The nature of statistical learning theory. Springer science & business media, 2013.
- [64] G.-F. Fan, S. Qing, H. Wang, W.-C. Hong, and H.-J. Li, “Support vector regression model based on empirical mode decomposition and auto regression for electric load forecasting,” Energies, vol. 6, no. 4, pp. 1887–1901, 2013.
- [65] T. Takagi and M. Sugeno, “Derivation of fuzzy control rules from human operator’s control actions,” in Proceedings of the IFAC symposium on fuzzy information, knowledge representation and decision analysis, vol. 6, pp. 55–60, sn, 1983.
- [66] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, Classification and regression trees. CRC press, 1984.
- [67] G. James, D. Witten, T. Hastie, and R. Tibshirani, An introduction to statistical learning, vol. 6. Springer, 2013.
- [68] S. Ruder, “An overview of gradient descent optimization algorithms.” Internet, 01 2016.
- [69] F. Galton, “Regression towards mediocrity in hereditary stature,” The Journal of the Anthropological Institute of Great Britain and Ireland, vol. 15, pp. 246–263, 1886.
- [70] A. Papoulis and S. U. Pillai, Probability, random variables, and stochastic processes. Tata McGraw-Hill Education, 2nd ed., 2002. Paragraph 5-4, pp. 139-152.

