Enhancing Wind Energy Management:

Predictive Modelling and Data Analysis Insights

Ari-Pekka Kujala

Bachelor’s thesis, AMK
May 2024
Information and Communication Technology

Jyväskylän ammattikorkeakoulu
University of Applied Sciences

Kujala, Ari-Pekka

Enhancing Wind Energy Management: Predictive Modelling and Data Analysis Insights

Jyväskylä: Jamk University of Applied Sciences, May 2024, 64 pages.

Bachelor of Engineering, Information and Communication Technology. Bachelor's thesis.

Permission for open access publication: Yes

Language of publication: English

Abstract

Wind power output prediction was researched to optimize energy management and sales in the renewable energy sector. The primary objective was to develop accurate predictive models for forecasting wind power output over the next 36 hours using machine learning techniques, benefiting turbine owner Tuuliveikot Oy by optimizing energy sales and management.

A comprehensive dataset from a wind turbine in Kauhava, including variables such as wind speed, nacelle direction, air temperature, and power output, was utilized. Data from the Norwegian Meteorological Institute (Met.no) was integrated into the predictive models. Implementation involved iterative development, including data cleaning, feature engineering, and the application of machine learning and neural network models to improve prediction accuracy.

The models developed demonstrated a mean absolute error of 300 kW for a 2,5 MW turbine, indicating that while the predictions were not perfectly accurate, they significantly reduced errors compared to simpler methods. Wind speed transformation was found to be crucial for enhancing model performance, and the models generated valuable insights despite challenges in data alignment and model complexity.

The research highlighted the complexities in wind power prediction and the importance of understanding underlying physical phenomena. The iterative process provided valuable insights that guided the refinement of methodologies. Although the models did not achieve perfect accuracy, they demonstrated the potential for significant improvements in wind power forecasting, contributing valuable tools for the renewable energy sector.

Keywords/tags (subjects)

wind energy, energy management, wind power forecasting, machine learning, neural networks, renewable energy, predictive modeling, data analysis

Content

Figures

Tables

Terms and abbreviations

DataFrame	**Data table in pandas library**
Met.no	Norwegian Meteorological Institute
NWP	Numerical Weather Prediction
MAE	Mean absolute error
MSE	Mean squared error
RMSE	Root mean squared error
PCA	Principal component analysis
SVR	Support Vector Regression
IQR	Interquartile range
R-squared	Coefficient of determination
FMI	Finnish Meteorological Institute

1 Introduction

1.1 Climate and Geopolitical Challenges

Due to global warming, a big shift towards green energy is needed. Annual global installations of wind power need to quintuple from 2020-2022 levels, or we won't reach the needed wind power to stay on the 1,5 Celsius temperature rise path by 2030. In 2022, the global wind capacity increased by 77,6 GW to a total of 906 GW, and the first TW (terawatt) limit is estimated to be broken by the end of 2023. (Hutchinson, Zhao et al., 2023, pg. 2)

Current world events provide good incentives to make the process for electricity better and more efficient. Wind power was increasing rapidly before, but the Russian attack on Ukraine in 2022 made it clear that energy ties to the east must be cut. This was good news for the green shift, as wind energy is one of the fastest and simplest ways to increase electricity capacity. Relatively small infrastructure needs and investments ensure a faster path to production.

According to the Finnish Wind Power Association, the number of wind turbines in Finland almost doubled from 700 to just under 1400 between 2018 and 2022. Their cumulative power capacity was 5677 MW, and they covered 14,1% of Finland's power consumption. By the year 2028, coverage is expected to reach 28%. (Press release, Finnish Wind Power Association 2023.)

The report of the year 2023 by the Finnish Wind Power Association states that there was a 23% increase in Finland's wind turbine capacity, and total production of the year was 14,4 TWh. (Finland's wind power statistics 2023, Press release, Finnish Wind Power Association 2024.)

1.2 Energy market

Electricity producers sell electricity in the same way in the market as other financial assets, with some differences. The general idea is still the same. Electricity is offered and bought in daily sales. The price is determined by estimated supply and demand. Therefore, it is quite important to have an idea of how much power the seller has at hand to offer for sale. If the produced wind power is lower than promised, the power deficit must be bought from Fingrid's balance service at a price that greatly reduces profits. Similarly, being able to sell an amount closer to actual production will bring better profits.

In Finland, wind power production is competing for the 2nd position after nuclear power with water and bio power. Energy prices don't directly follow wind power availability. In a closed market, this might be possible, but Finland is part of the European energy market, and energy is sold first to the direction where producers get the highest price. However, this is only half the truth because the capacity of power lines between greater areas creates bottlenecks for power transfers.

The need for this project comes from the way electricity is sold in the marketplace. Energy producers give their offer of energy availability and price in the afternoon for the next day. This creates a need for the best possible prediction of available power for the next 36 hours.

1.3 Project overview and specifications

This project aims to create value for the company by improving predictions for the next 36 hours of power output and conducting turbine data analysis. The Finnish Meteorological Institute (FMI) offers a solution for power and freezing predictions, indicating a need for such systems. Since FMI already provides power predictions, this project focuses more on data analysis than initially intended. Although power prediction is available from FMI, the wind industry might benefit from this analysis if the findings can be applied to other turbines and locations. By making better predictions easily available throughout the industry, wind power becomes more profitable, a better investment, and more enticing to produce. Even if the analysis results are applicable to a small portion of turbines globally, the benefits could still multiply to gigawatt levels.

In Finland, the possibilities for production predictions without a model like the one created in this project are limited. FMI does not directly provide 1-hour interval wind speed forecasts days ahead; for the next few hours, the forecast is given in integers. A difference of one meter per second in wind speed can mean as much as 700kW difference in power due to other variables such as air density.

Understanding the energy market and atmospheric physics are complex fields of study. To narrow down the project scope, only a basic understanding of the market is included. Understanding the physics behind the predictions is more valuable for this project than comprehending the financial aspects. The goal is to achieve good enough results for the model and find opportunities to increase production from data analysis. The initial assumption was that prediction is relatively simple, allowing for a focus on comparing traditional machine learning algorithms to neural networks. However, tuning hyperparameters for machine learning turned out to be time-consuming, leading to the use of a simple non-optimized linear regression as a baseline for comparison.

This thesis is part of the Resilience of Modern Value Chains in a Sustainable Energy System (Kestävän energiajärjestelmän modernien arvoketjujen resilienssi - KEMAR) project, co-funded by the European Union and supported by project's stakeholders and JAMK University of Applied Sciences. The order and data for this research came from Tuuliveikot Oy, with senior lecturer Mika Rantonen from JAMK University of Applied Sciences acting as the project commissioner. Because contact with Tuuliveikot Oy was limited, detailed specifications of the desired model, outcomes, and acceptable results were developed within the project.

The data provided by Tuuliveikot Oy comprises one year of 10-minute interval data from 2018, including records for wind speed, power output, air temperature, nacelle direction offset compared to wind direction, system state, and nacelle direction. Tuuliveikot Oy has one 2,5 MW turbine in Kauhava and had initiated a project to build more turbines. Noise and shadow studies identified planned positions for the new turbines (Heikkola, 2016). The project has since been sold to OX2, a company focused on green energy technologies, which is currently in the planning phase for seven new turbines (Tiedot hankkeesta, n.d.).

Having only one turbine in operation was advantageous for this project, as large wind farms can cause wake effects on turbines behind them, impacting efficiency. Predicting wind power for even one turbine is challenging, and the project's specifications required consideration of various modelling approaches. For a 36-hour prediction window, options include building a model to predict the total sum of the next 36 hours or an hour-interval wind power regression. These options serve as starting points for the model architecture. A 36-hour summed total could utilize Convolutional

Neural Networks (CNN) and Recurrent Neural Networks (RNN), with CNNs commonly used for image and visual pattern recognition and RNNs for time series predictions.

The project's purpose is to explore possibilities rather than present a final product. Productizing the model would be relatively simple, such as running it in a Python container that loads the latest information and sends predictions to a web page. The goal is to create a model that considers data availability if put into production, rather than achieving the best possible accuracy by using unrealistic data inputs.

2 Research methodology

2.1 Research description

This study focuses on predicting the power output of a specific wind turbine in Kauhava using machine learning techniques. The research utilizes one year's worth of comprehensive data provided by the turbine owner, encompassing variables such as wind speed, nacelle direction, nacelle deviation from wind direction, air temperature, and power output. The primary objective is to develop accurate predictive models capable of forecasting wind power output for the next 36 hours at any given time, with good precision. The data used to predict the power output is collected from the Norwegian Meteorological Institute through a loading script made by the writer.

The project involves several sequential steps. Firstly, the collected data goes through careful data analysis to understand more about the work ahead. After evaluating the need for processing the data, missing values and outliers are to be fixed. Additionally, feature engineering will be conducted to create new informative features that might help with the complex relationships within the data.

The project goes through multiple iterations to gain more knowledge about the phenomena and to achieve results for the finished product. The findings of this research are expected to contribute valuable insights to the field of renewable energy. The developed models will provide turbine owners with tools to optimize sales.

2.2 Research methods

This research is conducted in two parts. To get the most out of the applied research, the analysis of all involved data must be thorough. This analysis aims to find value from existing observation data. Outcomes of this research and development aspect could involve better nacelle alignment predictions or similar output-increasing activities. This combination of different research methodologies could be considered applied research and development.

The applied research part of the project collects information gathered through the analysis of observations and combines the knowledge into one working prototype model. Machine learning modeling uses ready libraries, and research on that subject is minimal. Research on machine learning consists mainly of testing algorithms or model architectures for the best accuracy. The model architecture is expected to be somewhat simple, as the physical phenomena are straightforward. The hypothesis is that the observation data can be predicted with forecast data. The forecast data is the same data meteorologists use for professional weather forecasting. The real question is whether the data collected for this project is good enough to get valuable predictions. The value created from the thesis is impossible to calculate, as that would require business knowledge from the turbine owner. Even simulating the value for history would need more information from 2018 sales, as the observation data only has turbine production, independent of the business.

Research and development for the problem at hand is difficult to plan before any analysis has been done, as only the exploratory data analysis will show what possibilities open with the data. FMI already provides freezing predictions with the wind power, so that is most likely seen in the turbine data as well. The data could show how much power is lost due to freezing, but that would need a model to provide predictions of expected power without too big an error. The data will be analyzed with charts and figures to visualize different correlations of data dimensions.

In the larger picture, knowing the scale of wind power production, even a small increase in power production would mean significant value worldwide.

2.3 Research questions

As the aim is to create working prototype for a predictive model, the applicative research in addition to the model consists of analysing the predictive power of the model. Model analysis aims to gain insights into accuracy and ways to improve the model with these questions:

What is the predictive accuracy of neural network models using Norwegian Meteorological Institute (met.no) forecast data in terms of key performance indicators such as R-squared, MSE and MAE?

- This question focuses on quantifying the effectiveness of neural networks in predicting wind turbine output. It seeks to define and measure the accuracy of the models using established statistical metrics. The analysis will include comparisons of predicted values against actual power output data, discussing the implications of the results for practical applications in wind power management.

How can model performance be optimized when predicting wind turbine power output using meteorological data?

- Explore different strategies for enhancing the performance of the neural network model. This could include experimenting with various network architectures, tuning hyperparameters, or integrating additional data features such as wind speed in gust and air humidity. The goal is to identify methods that significantly improve prediction accuracy and model robustness.

Analysis of wind turbine observation tries to find potential ways to increase efficiency of the turbine with following questions:

How do variations in wind speed, nacelle direction, and deviation angles impact the power output of a wind turbine?

- This question focuses on analysing the correlation and potential causation between wind characteristics and turbine efficiency. By segmenting the data into wind speed bins and analysing power output relative to nacelle direction and deviation angles, you can identify ways to modify surroundings of the turbine. The study will employ regression analysis or other statistical methods to quantify these relationships.

What patterns can be identified in the wind turbine data through visualization techniques, and how do these patterns correlate with environmental conditions such as temperature?

- Aim to uncover hidden patterns or anomalies in the data by using advanced visualization techniques (like scatter plots). Investigate how these patterns vary with changes in environmental conditions like temperature. This question would involve not just the presentation of data but also a discussion on the implications of these patterns for turbine operation and maintenance.

3 Foundations of Wind Power Forecasting

3.1 Domain knowledge

3.1.1 Wind physics fundamentals

This research aims to predict wind power using weather and power observations from the turbine, along with weather predictions from the Norwegian Meteorological Institute (met.no). Predicting power with the method used here does not require thorough wind research. Weather forecasting is already conducted by international projects, and this model uses those forecasts to determine the output power. Oversimplified, the wind speed forecast of X equals Y wind power. All work is done with machine learning algorithms, and the goal of the study is to find answers regarding whether it is possible to get reasonable results with this data, how it can be achieved, and how good the results are.

Wind power could be calculated mathematically if all the factors are known, but often some variables are hidden. “The fundamental equation of wind power answers the most basic quantitative question – how much energy is in the wind” (Kalmikov 2017, 3). He derives this equation (1) in his paper Wind Power Fundamentals from 2017.

$$\boldsymbol{E} = \frac{1}{2}\boldsymbol{A\rho v^3} \qquad \textbf{(1)}$$

In this formula, E is Wind power, A surface area of the propeller, ρ is air density and v is wind speed. With this wind power formula, one can only calculate the wind power density. This means the wind has a certain amount of energy, but multiple variables determine the actual wind power the turbine can extract. The roughness of the terrain, turbine capacity, turbine efficiency, turbine blade length, and wind direction play a significant role in the wind power gained through the turbine. Real turbine power calculation is also affected by the coefficient of performance, generator efficiency, and gearbox bearing efficiency. This coefficient can be calculated by dividing the wind power output by the wind power density (WPD) (Kalmikov 2017). As the formula shows, the cubic relationship of wind speed causes the power curve to rise slowly at the start and then rise rapidly after a certain point.

3.1.2 Boundary Layer dynamics

An openly available web calculator calculates wind speed at different roughness levels and heights. For example, in roughness class 2, the wind speed at 100 meters is 1,5 times the wind speed at 10 meters (Danish Wind Industry Association, 2003). According to maps, the surrounding areas are forest and fields, so the roughness can only be estimated by guessing.

Email exchange with Paavo Korpela, Meteorologist and Account Manager from the Finnish Meteorological Institute (FMI), revealed that while the wind speed is higher at 100 meters than at 10 meters, the actual phenomenon is called the boundary layer. The boundary layer of the earth and air causes the air to slow down near the surface. This layer also has turbulence and different flows that could be perpendicular to the wind direction. In three-dimensional space, these boundary layer turbulences also include vertical wind. The high positioning of the turbine tries to avoid these turbulences while reaching faster wind speeds. According to Korpela, seasons and time of day have a significant influence on the layer dynamics via the temperature vertical profile.

3.2 Turbine

Wind turbines measure wind speed at the top of the tower with an anemometer and wind direction with a wind vane. An early assumption was that wind turbines actively follow the wind, but investigations showed that the direction of the nacelle is adjusted by six electric motors. These motors consume electricity, so their use should be optimized considering the loss from the nacelle and wind direction difference. With this optimization, the nacelle can remain oriented in the same direction for long periods without significant losses in production. Predicting wind or nacelle direction is not within the scope of this research, but creating better direction optimization could be a topic for future projects.

Turbine data was delivered as an Excel file with documentation of the address and location coordinates. Explanations for different columns were provided with the Excel sheet. In most cases, the information was clear and thorough, but, for example, the state of the turbine only indicated that the productive status was 4.

According to the manufacturer of the turbine type used in this project, the efficiency of the generator is 97% in both full and partial load. This efficiency is presumed to include all different losses inside the turbine. The nominal power of the turbine is 2,5 MW, and the rotor diameter is 100 m. The turbine's cut-in wind speed is 2,5 m/s, and the cut-out wind speed is 25 m/s. The turbine has a variable blade angle, but this angle is not reported in the data, and the ice management system is optional and remains unknown for this particular turbine. This turbine system is delivered with four different heights, two for each tower type. (Lagerwey 2018, 1-2)

Exact information about the height can be found in the noise and shadow studies.

3.3 Met.no AI/DA-project

The training data for this project was collected in a separate data analysis project (AIDA). This data gathering project was conducted simultaneously with the thesis work, and some initial usability checks for the training data were performed in the other project. Training data for the thesis was loaded using a script from the Norwegian Meteorological Institute (Met.no). This data consists of multiple weather variables, from X- and Y-axis wind speeds to snow thickness. Each variable consists of 10 ensemble members for each hour. These ensembles create a probability window for prediction through mathematical models, which is a broad area of study not in focus for this work. In simple terms, the mean value of these 10 ensemble members would be the most likely candidate for the forecast. (Kujala 2024.)

The AIDA project and this thesis project often intersected, and to validate the data loading, the correct wind direction and wind speed were calculated. This calculation was primarily done because the turbine data has two different columns for directions. For data understanding, this was beneficial, but for the thesis needs, it was better to load separate X- and Y-axis data.

One issue that needed to be resolved was the format of the data. Met.no creates prediction files in 6-hour intervals, with data extending up to 60 hours into the future. This means that each hour entry will appear in the training data 10 times, with different values, and can be combined into one turbine power output value. After the initial data, as time progresses by 6 hours, new forecast values for the next 60 hours will be created. For example, after 6 hours have passed, the previously last value will be 54 hours into the future, but with a new forecast made, it will be 6 hours less into the future, reducing the uncertainty. As many hours pass, the same time value will eventually be only 6 hours into the future, and the forecast will be quite accurate, in principle. When designing a system for future predictions, one must consider the present. When the present becomes the past, the latest data will represent a real forecast of 60 hours into the future. The problem here is that if only the most accurate training data (always 6 hours old) is used, forecasting 60 hours into the future will have higher uncertainty than the training data. (Kujala 2024.)

One option to simplify the data would be to have only one data prediction for each hour. Handling and transforming the data would be easier since many pandas functions cause troubles with a datetime index when the index values are not unique. This would again cause problems with non-matching training and “production” data formats. For example, if the training data were loaded first, it would likely be the most probable for 6 hours, or always hours 0-23, while data coming from the latest load would still be 36 hours. Research for AIDA found that uncertainties are small for 36-hour forecasts, but it might be better to use the same data format that will be used for the predictions. (ibid.)

A link to FMI’s wind power prediction system was found only near the end of the AIDA project, and while this realization proves the original research question about the possibility of power predictions with this approach, it forced a search for a new point of view for this thesis. (ibid.)

Using simple predictions for temperatures in three locations, it was clear that this data can be used to predict wind power output in the thesis project as well. The AIDA project revealed some examples of errors that are likely seen in other projects using this data, and some of the problems that might be found in the models. Predicting temperature in different locations showed mean errors of around 0.8 degrees Celsius, which seemed good at that point. In general, bigger errors were encountered in temperatures between 0 and 1 degree Celsius than at greater temperatures. To tackle this issue, it would be possible to first categorize the values into zero power, power, and maximum power. Having these large errors in the low range could indicate underlying issues in the used neural network functions, especially given the simple architecture of the AIDA model. Temperature forecasts were used to predict temperature. (ibid.)

3.4 Loss functions and prediction errors

Choosing the right loss function is crucial because it determines how errors are evaluated and guides the model's learning process. Mean Squared Error (MSE) is one of the most common loss functions used in machine learning. MSE works by taking the average of the squared differences between the predicted and actual values.

The main reason for using MSE is that it penalizes larger errors more heavily than smaller ones. By squaring the errors, MSE ensures that the model focuses on reducing these larger errors, improving overall accuracy.

Root Mean Squared Error (RMSE) is similar to MSE, but it takes square root of average squared errors, and it is more interpretable because it provides error values in the same units as the original data, making it easier to understand how well the model is performing. When power value is in millions, average squared error of acceptable 400kW will become 160000000000 and scale is difficult to understand and read.

3.5 Tools

3.5.1 Coding framework

The primary coding environment used for this thesis was Python, facilitated by JupyterLab. JupyterLab provided an interactive interface that enabled efficient code development, testing, and debugging. The flexibility of Python, combined with the visualization and documentation capabilities of JupyterLab, made it an ideal choice for the iterative nature of machine learning model development.

3.5.2 ChatGPT

ChatGPT was utilized to aid in problem-solving and to generate ideas. When faced with coding challenges or conceptual roadblocks, prompts to ChatGPT often provided useful solutions and suggestions. This AI tool assisted in refining algorithms, debugging code, and exploring alternative approaches in the model development process. Often problems were solved in the process of prompt engineering. Another good use for ChatGPT was to get model templates.

3.5.3 Analytics

Pandas was essential for data manipulation and analysis. Pandas enabled efficient handling and preprocessing of dataset, which is crucial for training neural networks and machine learning models.

Scikit-learn was used for implementing various machine learning algorithms and for its many different tools for model evaluation, feature selection and data processing.

Keras and **TensorFlow** were the basis for neural network implementation. Keras, with Tensorflow, provided the tools for building, training, and evaluating deep learning models.

GPU calculations were essential for training neural network, especially in the beginning of the project. Training took hours to complete and speeding up the process with GPU saved many hours.

3.5.4 Visualization

Seaborn, Matplotlib and Plotly were all used to create visualizations for the project. Earlier parts of the project mainly used simple plots from Matplotlib, and later more complex (and better looking) plots were done with Seaborn. Plotly was only used for 3D plot, when fourth dimension was needed, three dimensions and a colour. This three-dimensional interactive figure made it possible to see even more complex relationships within the data. (See figure 1.)

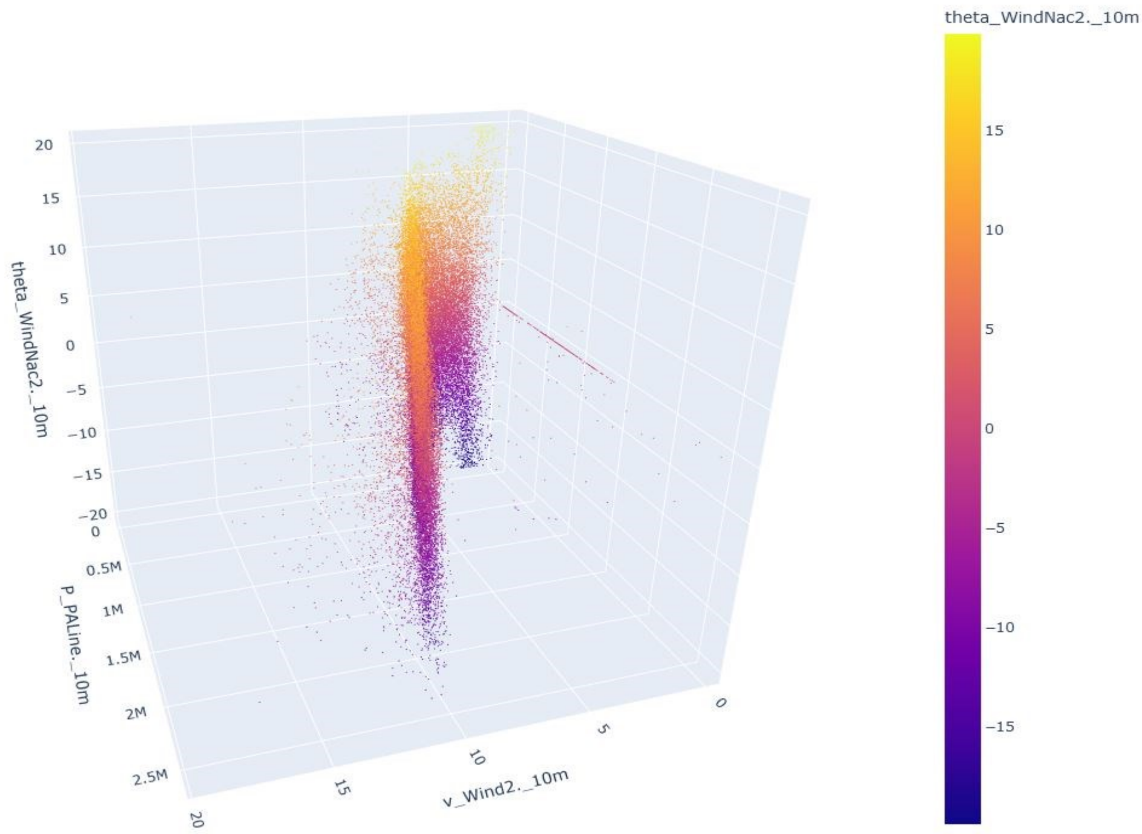

Figure 1. Three-dimensional scatterplot of the data made with Plotly.

4 Process

4.1 Iterative Development Process

4.1.1 Project Structure

Projects like this will usually follow a project framework like CRISP-DM (Cross-industry standard process for data mining). Expectation was that the process would be simple, and it could follow the process. Project will have quite clear structure of how the process will go forward, but while the project structure was not carefully planned beforehand, it evolved by itself to follow CRISP model. (See figure 2.)

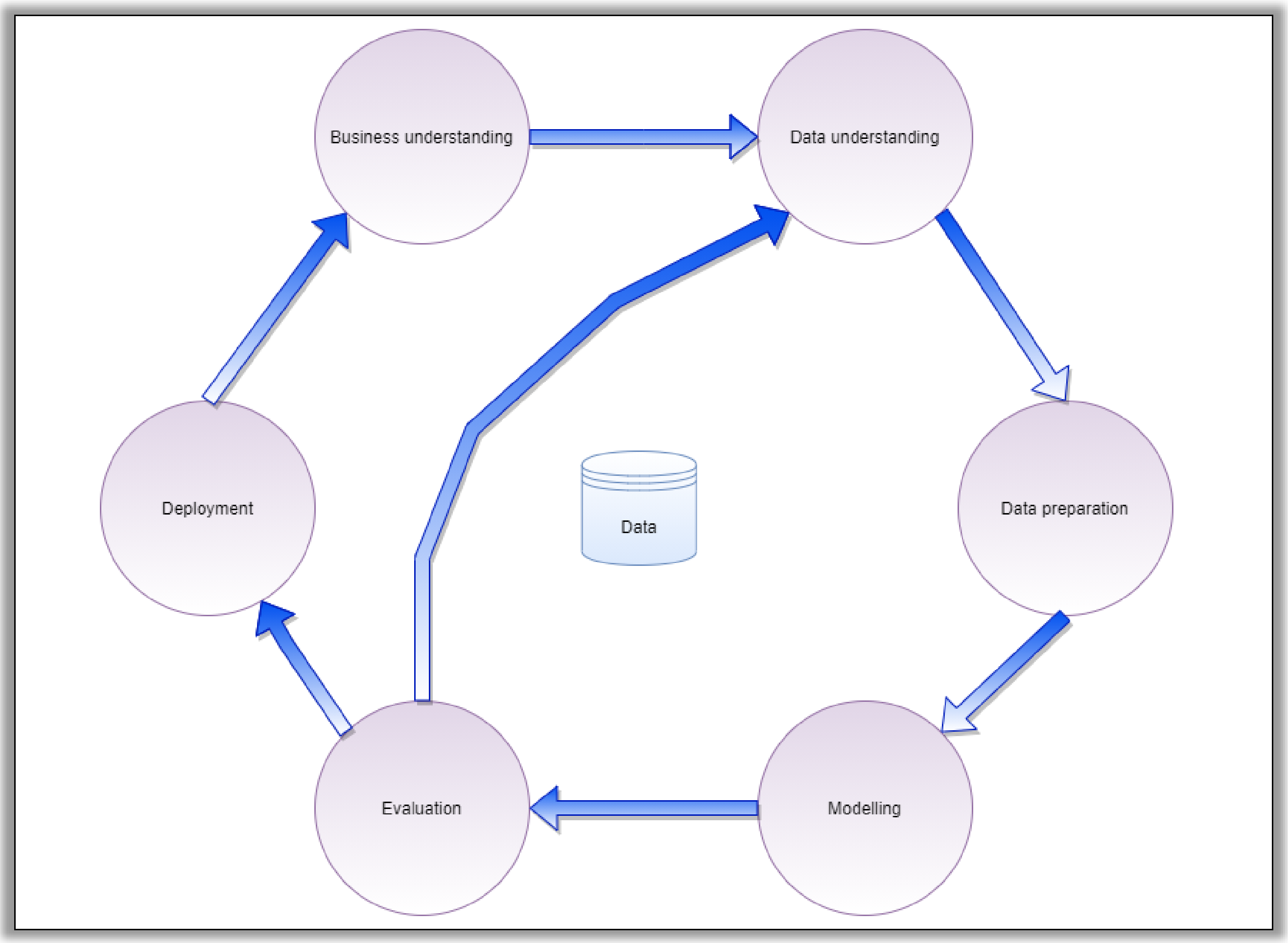

Figure 2. Crisp model cycle

First step to project like this, is to understand the business, and the value this kind of research might give. As it was understood from various electricity industry sources, the better predictions of the power output would reduce the need of buying electricity from fast reserve stock. Understanding how the value is formed, gives some indication of what kind of results would be needed. To be able to work with the data, it must be examined. What is included in the data, how do the values interact with each other and what is phenomena behind the data. Info page of the data had some explanations, like turbine is working normally if the ".state" column value is 4. Data had power and wind direction, and as it was understood, the wind direction is an offset value from nacelle direction and wind direction, that would mean that power of the wind in turbine would be as much higher than the offset would lower the amount of power the turbine gives out. Real world situation would work in three-dimensional directions but, as data from met.no gives x/y wind speed, same method is used to decide real wind power when direction is straight towards the wind. This means wind power in data is only the wind vector with offset of given direction angle. Idea behind this is pictured in Figure 3.

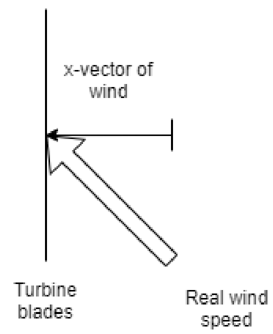

Figure 3. Wind direction and wind vector to against turbine

In this picture the example wind speed is 2,83 m/s and wind against the turbine direction at 45 degrees would be $cos(45°)×2{,}83\ m/s≈2\ m/s$.

4.1.2 Iterations

Quite early in project speed of the changes in turbine power output showed that hours are independent between adjacent hours. This was taken as a proof and justification for a decision to pick only one-hour regression to model. 1-hour independent regression is simpler to achieve as one row of data predicts one output value. In the time of the project, the electricity pricing was done by hourly estimates of consumption and production and with that it is important to predict the hourly power and not sum of 36 hours.

The iterative journey of this research began with an exploration of machine learning methodologies to understand the available data and the behaviour of variables. Initial iterations focused on employing machine learning techniques to establish a predictive model based on wind-related parameters. Grid search techniques were employed to experiment with various hyperparameters, a process that consumed considerable time due to the combined complexity for each additional parameter variation.

Subsequent iterations involved testing neural networks to assess their usability in predicting wind power output. However, the results yielded moderate improvements at best. In response, additional data, including gust, temperature, and humidity, was incorporated to enhance the predictive capabilities. These new features prompted the transformation of ensembles into mean, minimum, and maximum values, broadening the scope of analysis.

Despite these enhancements, the desired level of accuracy remained elusive. Therefore, careful examination of data distribution was conducted, revealing surprising trends in error distribution. Notably, lower errors were observed at specific power levels, prompting further investigation into the underlying causes and potential implications for model optimization.

Further iterations involved loading all data in 66-hour slices and investigating both training and testing datasets. However, the validation data exhibited significantly high errors, and scatterplot analysis revealed seemingly random predictions. To address this, the more variables from Met.no were loaded. However, discrepancies in data availability and quality posed challenges, implicating a need for preprocessing to fix anomalies and improve model performance.

As the iterative process progressed, various strategies were explored, including the possibility of creating separate models for different temperature ranges. The integration of additional variables, such as hour and month, aimed to refine model accuracy. However, despite these efforts, validation results remained inconsistent, prompting a shift towards a deeper understanding of the underlying physical phenomena driving wind power dynamics.

In later iterations, the focus shifted towards simplifying model architectures while incorporating custom loss and activation functions based on known physical formulas. Additionally, explorations into data alignment and correlation analysis aimed to enhance model interpretability and predictive accuracy.

Despite iterative refinements, challenges persisted in aligning model predictions with observed data. Experimentation with basic models, incorporating multiple outputs to differentiate power levels, frozen conditions, and other categorical variable, marked a shift towards a more pragmatic approach to model development.

In conclusion, the iterative development process highlighted the complexities inherent in predicting wind power output accurately. Each iteration contributed valuable insights and lessons, guiding the refinement of methodologies and approaches towards achieving the ultimate goal of developing good predictive models for wind power forecasting.

4.2 Known problems

Wind power prediction using machine learning methodologies presents multiple obstacles, each demanding careful consideration and resolution. These challenges, ranging from data compatibility issues to the complexities of model development and validation, underscore the intricate nature of predicting wind power output with precision.

The first problem identified at a very early stage was the disparity between old data and new data from Met.no. Forecast data from 2018 had maximum of 10 ensemble members, but later the amount changed to 30. This causes problems for productizing the model since the input data must be of the same shape. Earlier AIDA project had proven that the precision does not noticeably suffer from using values like mean, maximum, and minimum to transform the data into more generic form. This approach would also address possible missing values in the forecast data.

Weather forecast data from Met.no is averaged over a 2,5km by 2,5km grid, so exact forecast data is not available. A grid of 2,5km is large enough for the wind to take decent time from side to side.

Forecasting wind power from weather forecast produces errors that double in the process. When weather forecasts have errors of 1 m/s, and these errors multiply with each variable used, the worst-case scenario could result in unusable models, or at least errors so large that it would be the same as using readily available weather data and guessing.

In AIDA-project, the MAE for wind speed prediction was from 0,7 m/s to over 1 m/s. This was achieved by using wind speed ensembles and Finnish Meteorological Institutions observations and using a neural network to predict wind speed at the same elevation and location. The correlation of these 2 simple data rows is around 0,9 and correlation of FMI observations and turbine observations are only about 0,65. This low correlation of two observations from near locations imply that there is some unknown factor in effect.

Wind speed at a turbine height about 130 m is easily calculated using formulas found in different sources. Email exchange with FMI's Paavo Korpela, Renewable Energy Account Manager and Meteorologist revealed that a phenomenon called "boundary layer" plays major role in wind speed at heights. This could become one of the main issues affecting prediction accuracy.

Connection to original commissioner, Tuuliveikot Oy, is not available, so no new information from their side is expected. Questions about the observation data and how it is formed must be answered from the data if possible. On the other hand, deeper investigation of Met.no data is also out of scope for this project. New data can be loaded with the ready-made script, but possible new dimensions in that data will not be installed. As stakeholder's opinions or advice is missing, the limit for "good enough" prediction precision must be invented in the project.

Modelling projects often use naïve metric to get the maximum level of the error. This naïve way could be explained easily with binary problem, where half of the data is 0 and another half is 1. When all predictions are 1 the accuracy is 50%. This is even better when the data is not symmetric, when only 10% of the data is 1, having all set to 1 will result accuracy of 90% even if the model could not predict anything. In regression problem like this project, wind speed mean could be taken and from that the theoretical power could be calculated. This could be the baseline accuracy, which would not take the height difference into account and resulting wind power prediction would be in most cases far from truth. Discussion with FMI lead to understanding that the R-squared errors on FMI service are around 0,8 and that would serve well as a target. Results are not expected to get close to that since it would require more thorough knowledge about physical phenomenon, which is not in scope of the project, and more accurate data that is produced within FMI.

4.3 Data analysis

4.3.1 Basic analysis and cleaning

Before thorough analysis could be done, some general information of the data had to be gathered. This could have been done within the Excel the data was delivered, but pandas was better option, since much of the data work continued straight from the analysis. Any data errors had to be identified and possibly fixed.

First attempts to combine any met.no data to turbine data caused time index errors, which meant that one or both data were missing date times. After creating full one-year index and combining the data there, showed that there were multiple rows in the turbine data that had time couple seconds off. This was fixed by rounding every row to the nearest 10min. These seemingly missing rows were missed in early iterations since missing values were interpolated. Error was fixed when the interpolation was removed.

As it was known, the wind speed did not correlate 100% with the power, wind speed was used to calculate the theoretical power and plotted with the power output in Figure 4.

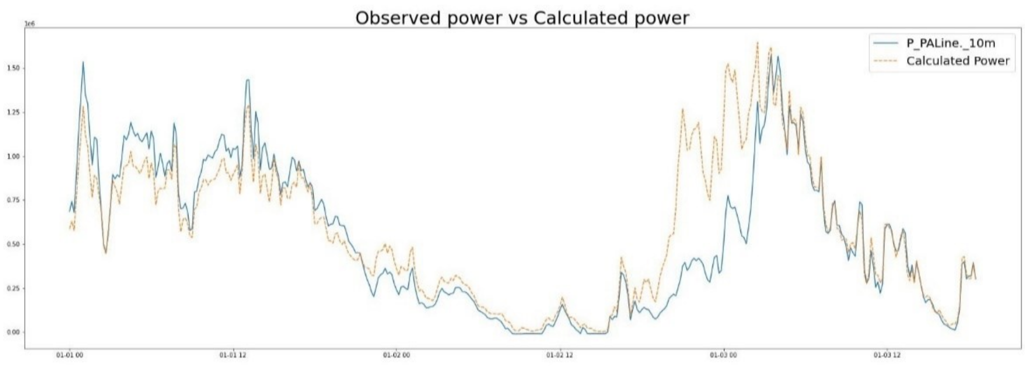

Figure 4. Observed wind power compared to power calculated from wind speed.

Wind power for this turbine could not be directly calculated with the wind power formula in section 3.1.1 since the turbine has maximum power output of 2,5MW. Wind formula has an air density as one variable, and because it is complicated to calculate accurately, often an average of 1,225 is used. The calculated power also did not include direction error of the turbine as anemometer is designed to measure wind speed, without directional aspect. Wind turbine efficiency plays a part in the power, and to get comparable values for calculations, the efficiency was needed. The efficiency changes depending on the speed so as this calculation used a mean value, the result is not very good, but it shows some information, and gives ideas of the causes. When the values differ a lot, either the speed or the power has an issue. For example, when the calculated power is larger than the power observation, it could indicate that the nacelle direction is off or the turbine is frozen, and when the observed power is higher than the calculation, the anemometer might be frozen. Usually, the latter was because of air density being lower than used average 1,225, since when the anemometer got frozen it shows zero wind speed.

Air density is a big factor in the formula, a small test was done to see how the density can be seen in the power curve of the theoretical power for a turbine of this size. (See figure 5.)

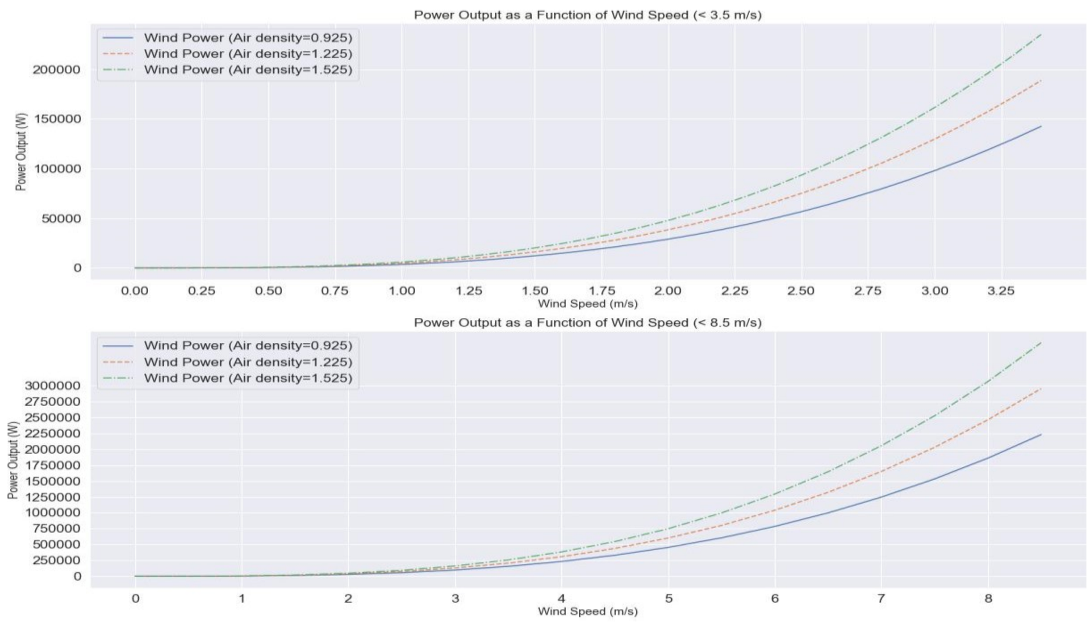

Figure 5. Air density effect in power curve.

This curve also shows how fast the curve rises every 1 m/s. Upper picture was used to understand the curve around 2,5 m/s area that is supposed to be the lower limit of the turbine. Three separate curves show that the difference between different densities can be rather large. At 2,75 m/s wind the power at 1,225 density is 100kW and ~25% increase to the density creates about 125kW. This is seen in the formula already, but visualization helps to understand scale of the effect of a small change. The figure does not directly transfer into the turbine power, as turbine efficiency is around 40-50% at best.

4.3.2 Analysing turbine data

In the start of the project, before anything was done, the way the model was going to be done had to be decided. An initial idea and presumption were that some kind of time series model must be done. Solving the fundamental logic with the model was even bigger issue when interval of 10 minutes was planned. Distinction between a weather forecast and a turbine power forecast was logically narrow. Data showed that the turbine follows the wind speed directly, as much as power in 10-minute interval changes. Real testing of the speed would need to have bigger differences between data rows, but as the wind in most cases do not have 20 m/s speed on one minute and 0 after 10 minutes. This test used the calculated power to see how fast the calculated power moves compared to the observation. It was quite safe to say that if the reaction speed is so close in 10-minute data, in one-hour intervals it is a valid presumption that direct regression of per row is good approach.

Real analysis of the turbine data started with various plots. Turbines wind power curve gave several insights of the data. (See figure 6.)

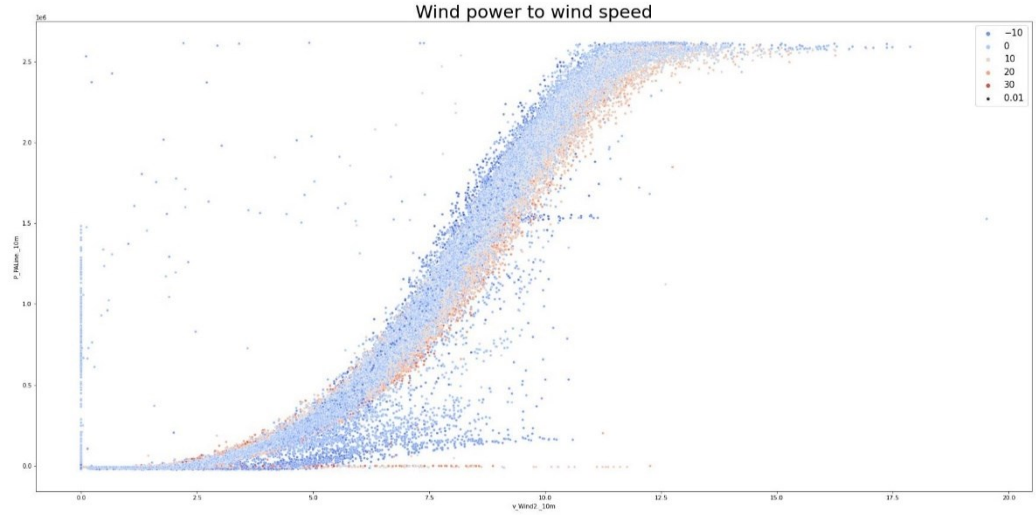

Figure 6. Power curve of the turbine with air temperature coloring.

In this figure, you could see that there are many datapoints with very low power output for the wind speed. They all are on negative temperature, which would suggest that the turbine has frozen. Many power readings can be seen with zero wind, so these are data points where most likely the anemometer has frozen. Some of the zero power rows have notable wind, which might be indication of the state of the turbine is not operational. Thickness of the curve tells that only a wind speed cannot be used for the power prediction. One notable information gained from this figure, is the difference between cold and warm sides of the power curve. Only general idea can be used since many red points are left under the blues. Most of the points under the curve are on negative temperatures, and this is enough to assume that the turbine does not have the optional anti-freeze equipment installed. This data could be used to calculate how much power was lost in a year due to freezing. Calculating the lost power directly from the wind cannot be done since the wind anemometer also gets frozen. As possibility to freezing is left with the turbine, this cannot be used as an outlier or an error in the data, but a real phenomenon that can happen in the future.

These frozen rows would in future cause another problem for validation. Initial plan was to create a model with data from January to mid-December and do testing with data from latter half of December, but most of the frozen data was from that to-be-test data. With more years in data this could be done in a way that even the frozen times could be calculated correctly but for this data it was impossible. In the end the data from summer was used as a “new unseen” data to avoid problems with frozen data. Hopefully the model can catch at least some freezing points, but there is no way to validate this without new observations.

Freezing machines could also be a reason why wind speed and wind power had such low correlation, even when they are supposed to be directly related. The correlation of wind speed and power was around 91% at start, removing the non-functional states and fixing some other errors got the correlation to ~97%. These are only observations and could only be used for training model from turbine wind speed to power. One of the most valuable insights was that the wind speed and the wind power could not be used directly as a truth. This would have most likely caused problems in the approach that was planned for machine learning, to first use the wind speed as a target variable and then predict power with that. Also, this would not have been very accurate since as we already know, the wind speed is not the only factor in the power output.

To find out a way to see the errors in the data beyond “.state” value and frozen data, it would be important to have an accurate prediction of the wind power at hand to distinguish between good data and error in production, but at this point this was not acquired yet. To create a workaround for data cleaning a calculated power was needed. This calculation needed an efficiency of the turbine. Only efficiency value from turbine manufacturer was 97% and this could only be efficiency of the turbine machinery.

Basic analysis revealed that the manufacturer specifications are not a rule for the power limits. The cut in speed of 2,5 m/s is perhaps not activated, as it seems that curve starts to rise before 2 m/s winds. Zero power, in fact means negative power, as the turbine itself consumes some power, minimum power being around negative 25kW. The upper end of the power curve does not stop at 2,5MW but closer to 2,6MW. The curve starts to even out after 2,5MW which was interpreted as some kind of braking system, most likely adjustable blade angle, being used for this turbine.

Efficiency is a factor of how much power a turbine can extract from theoretical wind power. And to validate this logic efficiency was investigated with wind speed. (See figure 7.)

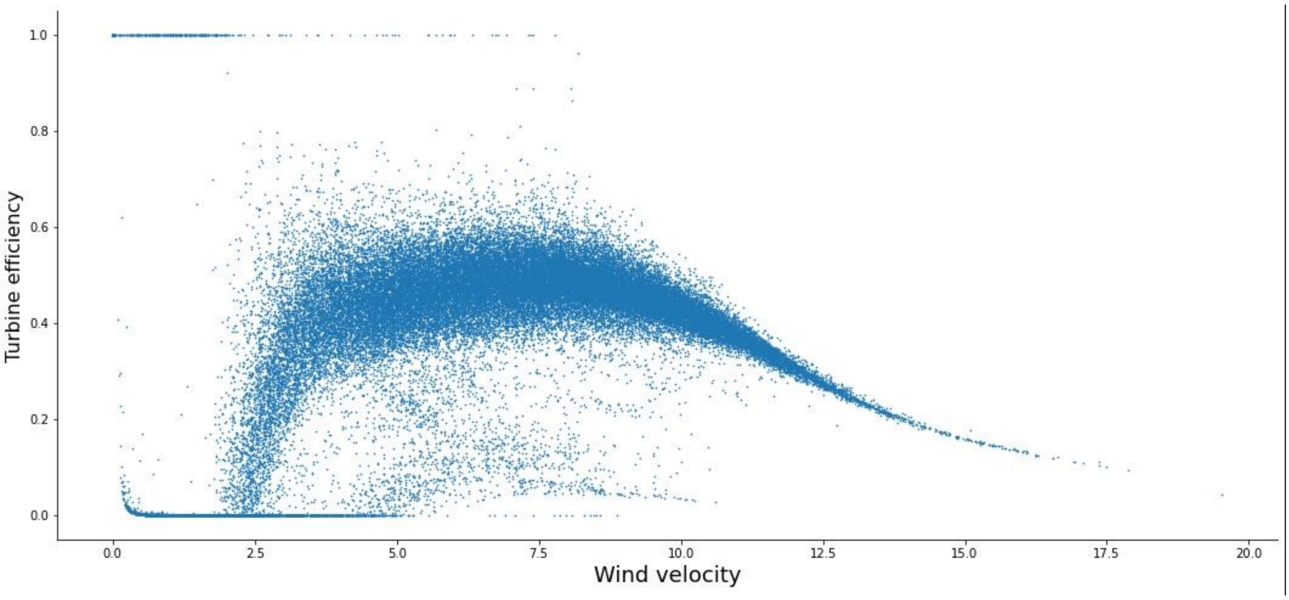

Figure 7. Wind velocity compared to turbine efficiency.

Efficiency maximum can be seen quite well between 3 m/s and 10 m/s winds. End of the curve is cut because of how the maximum power was clipped into the data. Data in this picture also shows many values that seem to be off. Low efficiencies are most likely another sign of freezing. For calculated power a mean value of this efficiency was used, and this figure shows that in most cases the mean works rather well.

At this point big part of the focus was on data cleaning to remove the values that do not hit the theoretical power curve. Importantly as the cleaning was done manually, this would not work in production, and this had to be solved. Either the "errors" should be labelled programmatically or left in the data.

Next investigation was conducted to determine how the nacelle direction worked in the data. According to info sheet in turbine data, the wind direction was supposed to be the real wind direction, but after plotting the values, it was clear that was not the case. (See figure 8.)

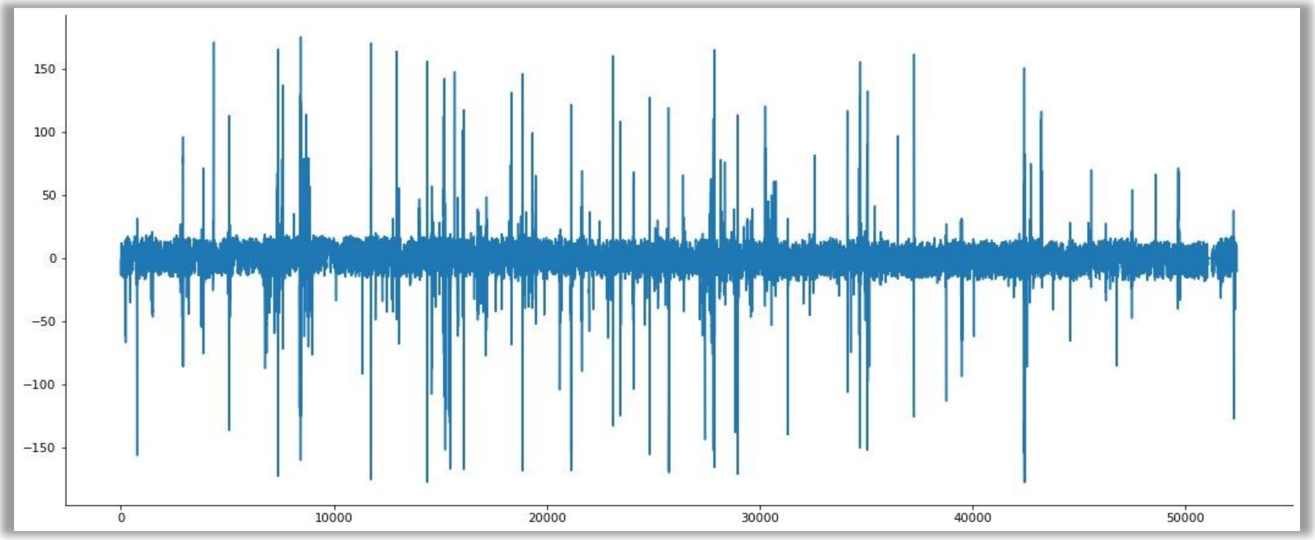

Figure 8. Wind directions in original turbine data.

Most of the values were around 0 and mean value was -0,0074, it was presumed to be the angle difference of actual wind and nacelle direction. Later when the picture of the turbine was examined better, the reality was easily understood. The anemometer that had measured the values, is on the tower, and will turn as the nacelle turns, so original assumption was correct.

One pressing issue with the turbine data is the actual power output that was main reason this research was committed. If a turbine measures 1MW of power, it means that with current nacelle deviation, the turbine can extract this amount. Basic understanding is that if the nacelle points 90 degrees towards wrong direction, the power would be zero. So, when we use these values as a target, this target is not the maximum, but the value that is dependent of the deviation. To insert the deviation as an input data, would mean that we would need to know the deviation 36 hours to the future. This creates a very difficult issue. To predict the future power, we could just predict the maximum power, and not take direction error into account. This would not be optimal way, but we know that we are trying to predict a value that has an unknown factor.

Quick analysis of the direction of the nacelle showed that the nacelle does not follow the wind but change with an unknown logic. The power output should be fixed, or a workaround would be needed.

Fixing the output would need a multiplier, how much does the deviation angle cause the power to decrease. An initial presumption was that when the deviation is zero, the power is at maximum and the power would decrease more when the deviation got bigger, towards the 90 degrees where multiplier would be zero. The data was divided into 1 m/s bins and a separate regression line was plotted for negative and positive values. (See figure 9.)

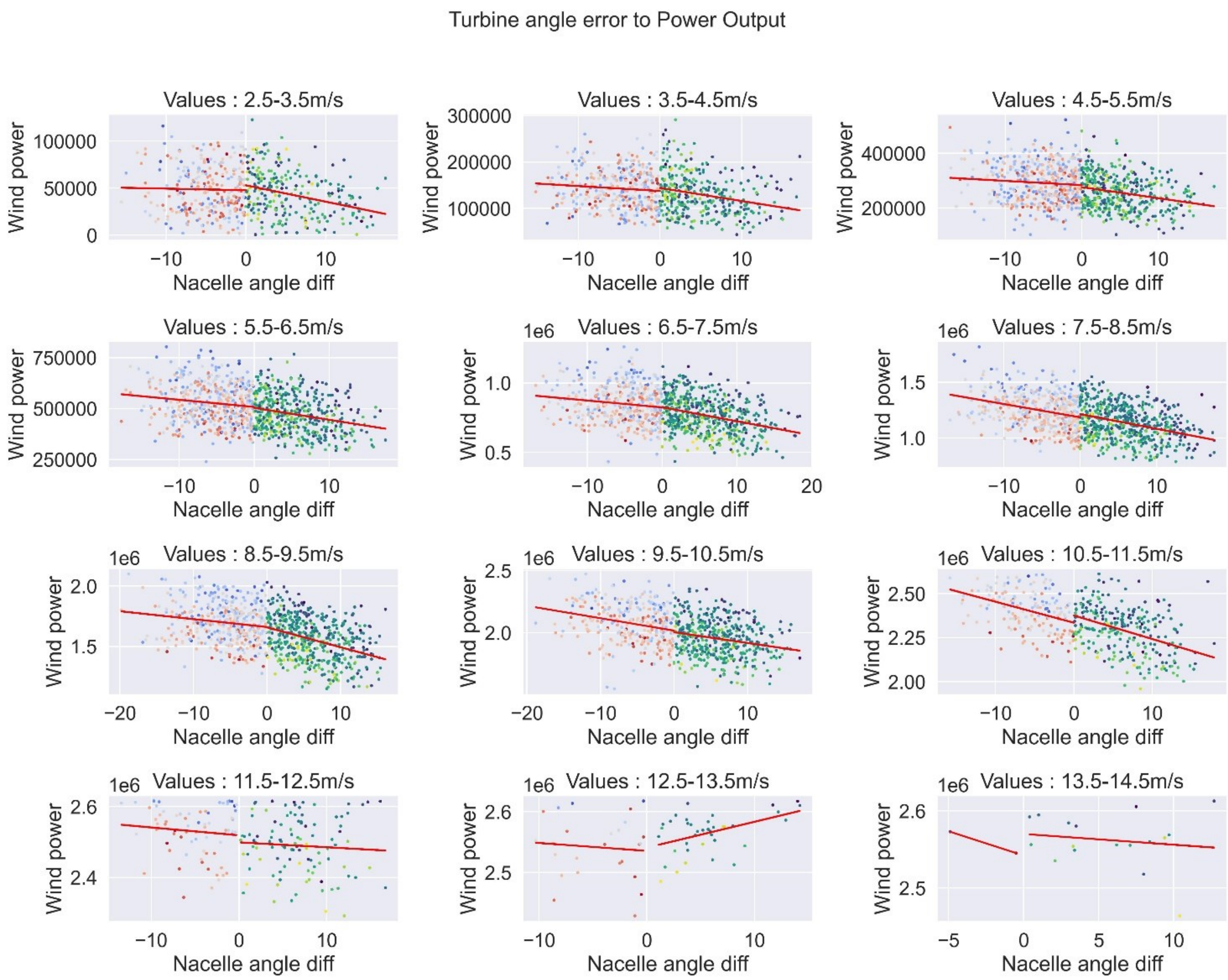

Figure 9. Power to nacelle error.

The lowest row can be ignored due to relatively low number of observations and the turbine being at maximum power. Colouring in the plot is coming from temperature. Results of the picture were not what was expected. Seemingly the turbine produces more power for basically every bin the more the deviation is to the negative side. This can be caused by either some unknown factor in turbine physics, wrong logic in the code or data issue. This figure has frozen values removed which were clearly seen below the main group, on cool colour. Test was also done to find out if for some reason the wind speed would not be equal on each side of the zero deviation. If the figure is right, this type of turbine might have better output with slight nacelle deviation to negative direction. One possibility is that the wind direction measurements are not good. Turbine blades could cause error in the wind vane used to measure the wind direction. Blade alignment and direction of rotation might explain the phenomena as well.

As this investigation was not a focus of the research, it was just noted that more work is needed on the matter, but there is a possibility for power increase of even 10%.

This finding means that with this knowledge, the power output cannot be corrected with simple trigonometry, which was used in early iterations to calculate corrected power output. Inability to calibrate the target value would cause problems with model prediction testing, and more in production, where we do not know the error. With this, the deviation angle was added to the training data, and predictions would be done with inserting zero angle deviation into the input data. Best way to test this, would be to have corrected output with zero angle deviation, but as currently it was impossible to fix the power, the validation would be done with the observed power and the deviation. One observation from tested “corrected power” was that when wind anemometer was frozen, this nacelle deviation could be very high, and cause very small dividers, which in turn would cause extremely high expected power, which obviously broke a lot of systems before this problem was understood.

As the AIDA project found out, there was notable errors in predicting wind of a point in same altitude. To find out the power difference inside the power curve, a rolling window of data points had to be created. The wind data was reported with one decimal point, and this was good enough for the work, but binning everything for 0,1 m/s intervals would cause outliers when there would be very few observations. To tackle this, combined rolling window and average was used to get somewhat reasonable curve. (See figure 10.)

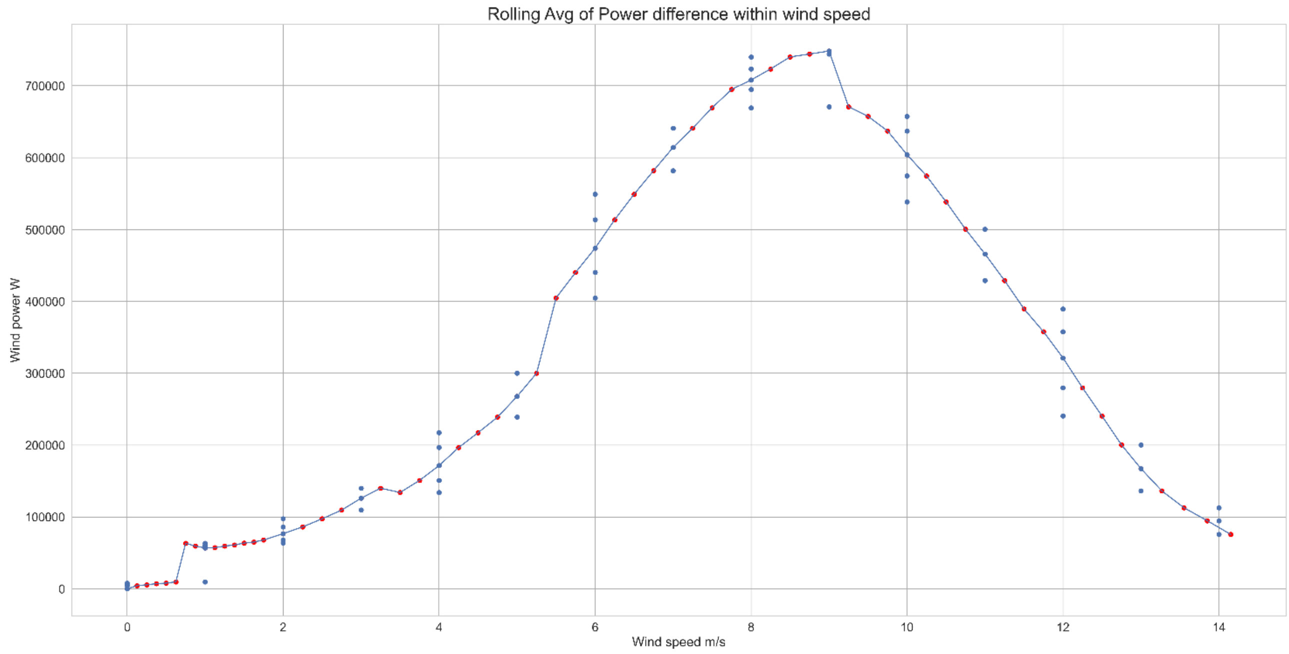

Figure 10. Curve thickness for wind speeds.

As we can see, for example 8,0 m/s can have 700kW difference in output. This could be used to validate an error level in future model. If the error can be ±350kW inside single value, then everything lower than this error could be considered as a good value. Even more when the weather forecast generally gives only 1 m/s precision. Some of the difference can be explained with air density but perhaps not all. Analysis for this was not done, as the reason was not important, and hope was that the neural network could find a way to go around it.

To gain any knowledge, it is important to understand the directions. According to the specifications given with the turbine data, nacelle direction is degrees from north, so 0 is to north. Nacelle deviation of 5 degrees most likely means that the vane shows wind to be on +5 degrees. So, the nacelle being straight to north, and the vane showing deviation of 5 degrees, actual wind direction would be 5 degrees from north.

4.4 Met.no data

4.4.1 Load script

Before the work with the data could be started, the script for training data had to be changed to suit the needs of this project. Initial testing was done simply using wind speed and wind direction.

All 10 ensembles were loaded, and already in the loading process the speed and the direction were calculated. Calculating anything already at the load phase was a mistake as the loading process was very slow and all calculations would be faster to do for the readily loaded data. First attempts were made with only 6-hour slices combined into one time series data, but as the production data would be only one 36-hour slice, the data had to be reloaded. Every file (4 times a day) had to be created as separate dataframe. To create one dataframe from over 1000 dataframes needed a lot of work and almost every iteration had different needs for data model. Along the way more variables were needed and adding new loads was easier when the script was working well.

4.4.2 Analyzing met.no forecast data

Analyzing the ensemble data was expected to be more straightforward than the turbine observation analysis. The analysis started by simply describing the data. Some faulty rows existed with very low temperatures or very high winds. These could be removed or left in, errors might happen later, so they were left in the data. Most of these were in the ensemble 9 which is supposed to be the most random. Nature of the ensemble model is that the different ensembles form a window of probabilities inside the ensembles. In the AIDA project it was noted that as often as 50% time the observations did not fit in the ensemble window, and surprisingly it seemed that neural network could predict better the observations that were outside the ensemble forecast. The ensemble forecast behavior against turbine wind speed can be seen in Figure 11.

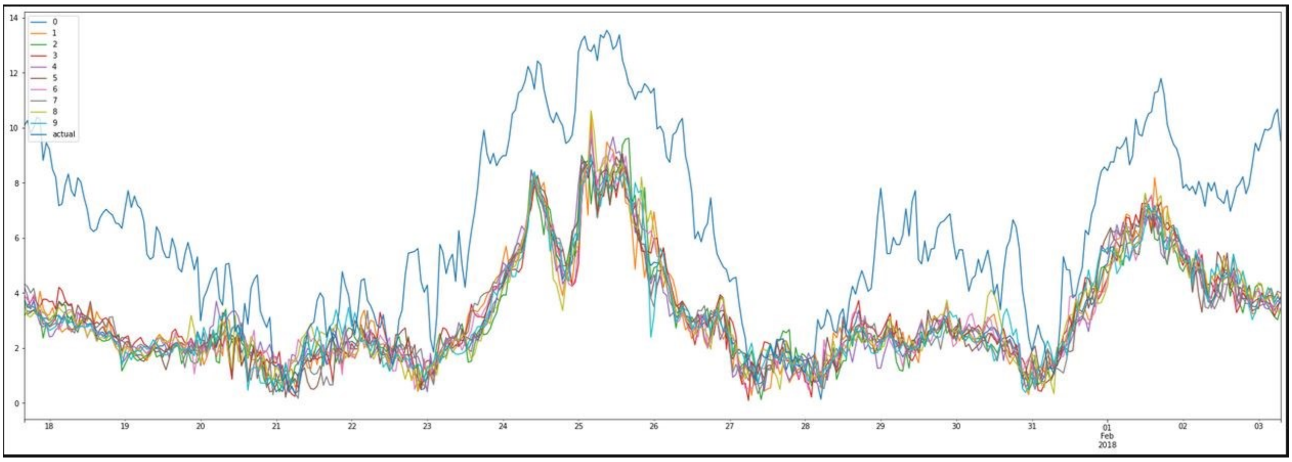

Figure 11. Wind speed ensembles and turbine wind speed.

This picture shows how the wind speed at ground level and turbine level differs. Ensemble data analysis showed that the ensemble 0 is the ensemble with least amount of randomness in the data, which means that the ensemble 0 correlates best, but as the nature of ensembles seem to jump from side to side, it was yet to be seen how the seemingly random motion affects the predictions. To simplify, if observed wind would be 10 m/s for minutes 0, 10 and 20, the ensemble 0 could be 10, 8, 12 and in average still be on point.

How this window can be installed to a neural network so it is seen as a window by the model and not individual variable, or if it matters, will be seen when the model training starts.

First loads had direction and speed calculated directly in the load script, research for the AIDA project noticed that wind direction needed an alpha calculation when it was projected on a globe. Calculation script for this alpha was a ready-made Python code by the Norwegian Meteorological Institute, and it calculates local grid rotation for the x and y axis (Norwegian Meteorological Institute). Also, the direct x and y vector calculated direction would give a wrong starting point and a different axis, this is seen in the figure X from AIDA. The Figure 12 (taken from the AIDA project) shows a wind direction time series for various ensembles, where the wind direction is measured in angles. The discontinuities in the wind direction are clearly visible.

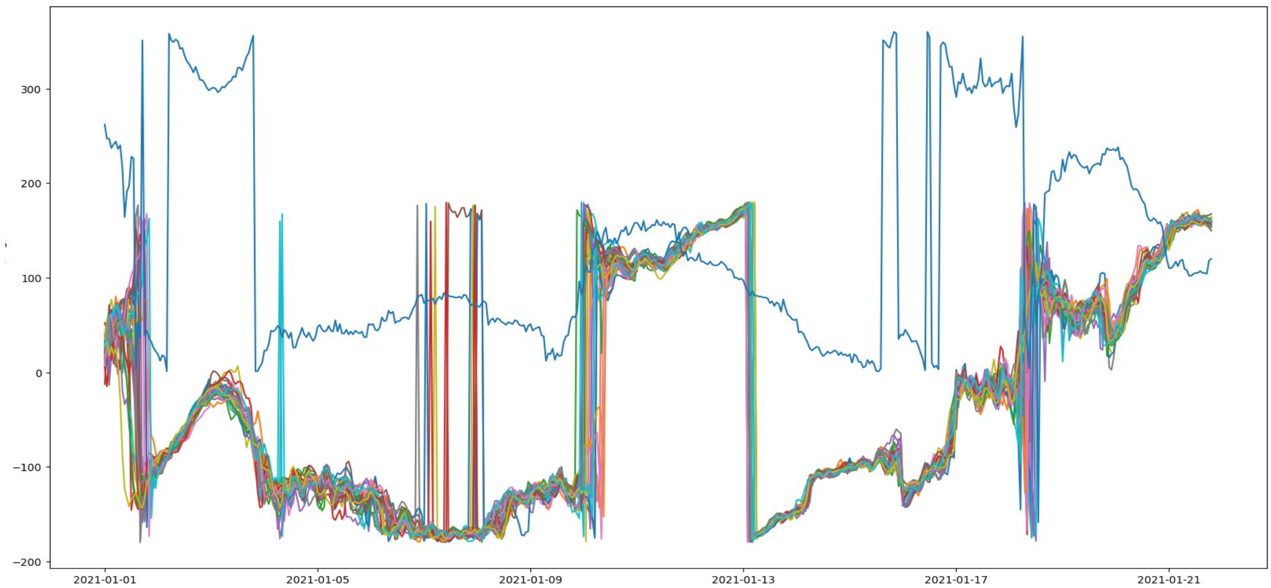

Figure 12. Wind direction calculated from x/y vectors compared to FMI wind observation. (AIDA project, Ari-Pekka Kujala)

Like this figure shows, the values are mirrored, off about 150 degrees, and one is 0-360 and the other -180 to 180 degrees. This visualization raised a suspicion that directly using angles would not work either, since 0 and 359 have one degree difference which would cause issues when interpolation was used. And later it was understood that the direction was not even needed for the training data, and even for the observations, wind direction would not provide easy solution for any of the problems.

As it was known there are problems with the correlation of wind speed observation and the ensemble, it was imperative that the ensemble load was working right with the coordinates. The script had multiple phases in it and any of them could have an error which would result in more errors later. Location was validated by drawing the whole map worth of grids and lowering temperatures of nearest grids to get the location showing in the map. (See figure 13.)

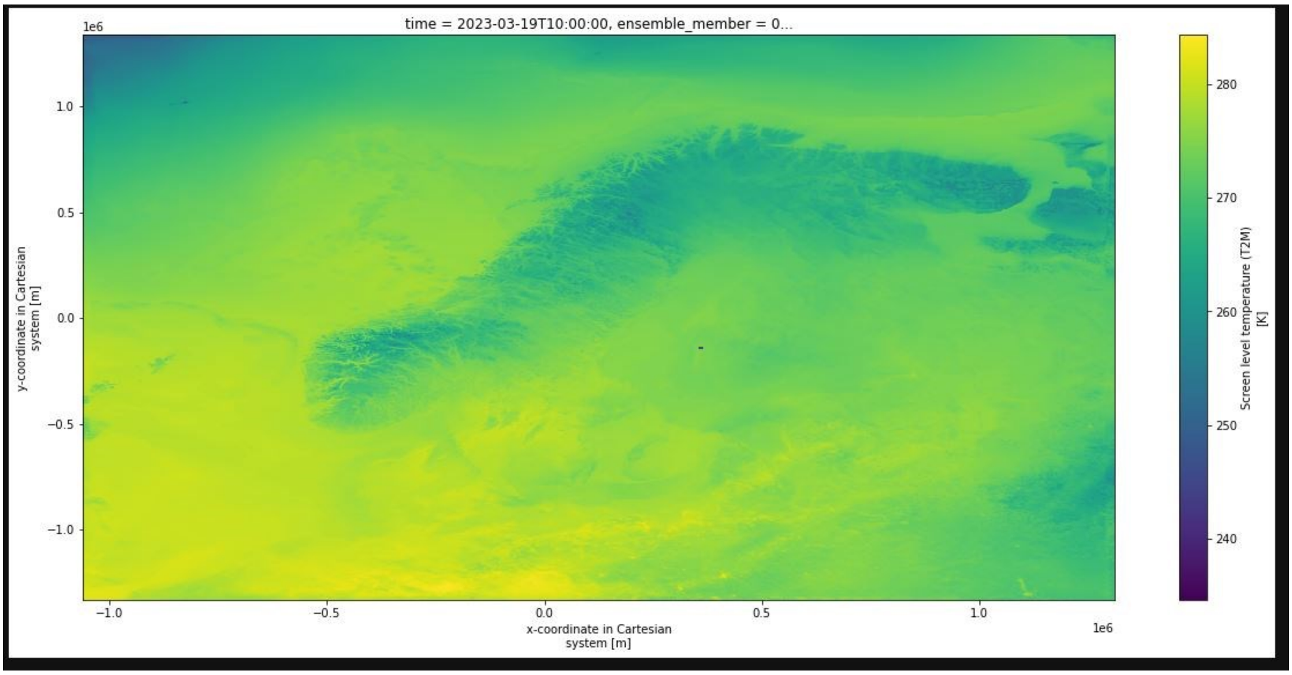

Figure 13. Easy way to validate a location.

The location was not easy to pinpoint to the given coordinates, but it looks close enough.

4.4.3 Data concatenation

The turbine data specifications did not state the time zone of the time index, so it had to be checked. Checking only by data if two different datasets connect to each other was not easy task since the data was not direct fit in any case. Usually, it would be reasonable to have the UTC time zone on machinery, but as the turbine delivers data to be used in Finland, it was possible that the data is in Finnish time. Plotting and correlation of temperature felt easiest options to start. Figure 14 shows the plotted temperatures for observations and 0 and 2 hour shifted ensembles.

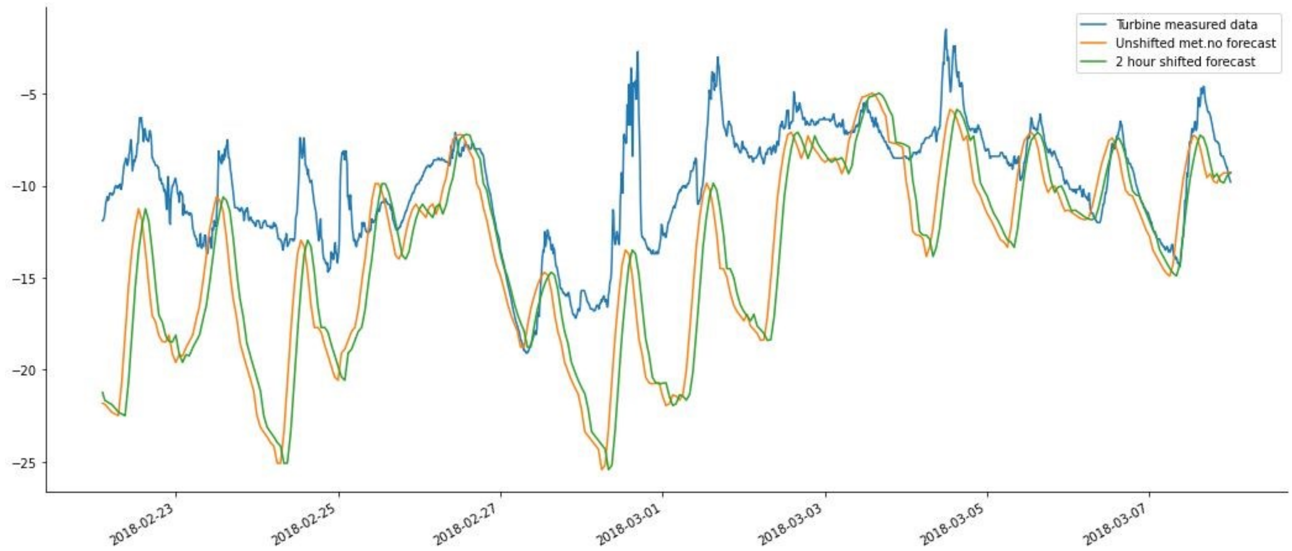

Figure 14. Zero and two hour shifted temperatures compared to weather observations from turbine.

As seen in the picture, it is very difficult to confirm the used time zone. To get some reasonable numerical answer to this, correlation of different time shifts was taken, and results are shown in a chart below.

Table 1. Correlation between observations and shifted temperature data.

Unshifted met.no forecast	0.965012
2-hour shifted forecast	0.969328
3-hour shifted forecast	0.967156
4-hour shifted forecast	0.962372

As seen in the chart, difference in correlation is not big, at best an indication towards the time zone used. As the difference was small and it was leaning towards UTC+2, combined with hints from the plot this was enough to have an educated guess it to be the case.

When forecast data was combined with the turbine observations, the big task was to figure out a reason why correlations between forecast and observations were so low. Correlation of 0,6 could cause big errors in the future. Also, turbine data was with 10-minute interval and ensembles were 1-hour intervals. To get more data into use, the missing rows were interpolated forwards so every 10-minute row would get training data with 1-hour change divided to 5 new values.

Another issue with data compatibility was that the ensembles were forecast for a point, and turbine data was average of 10 minutes. When connecting even hours 10-minute average to ensemble, the combination would be close, but still not full match. And from different angle, when turbine data is 6 averages of 10 minutes, the predicted value does not match what was needed. So, it was important to calculate the 1-hour average power output, so we know the real target variable. Taking average blindly would cause errors for nacelle directions, but only nacelle deviation was used, which should work as average. Similarly, the forecast data had to be transformed into something that might have biggest resemblance to 1-hour average, average of hour 0 and hour 1. This would cause some work for production side, since to get a prediction of hour 0, data for hour -1 would be needed.

This 1-hour data made the dataset smaller which sped up the model training. Having smaller data also made it easier to distinguish data behavior from the Figure 15.

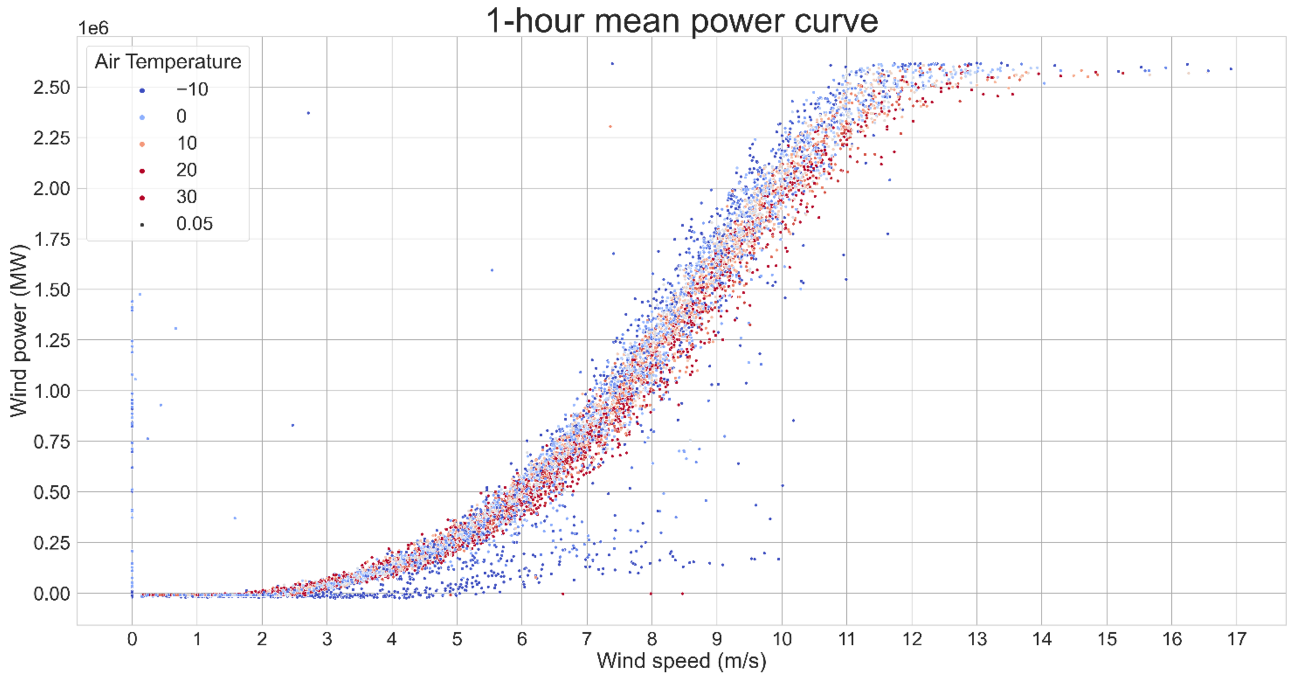

Figure 15. 1-hour interval power curve data.

Comparing the picture to the earlier full data shows that there are not many hours that have been frozen. Most of the data points under the curve are blue, which would indicate some level of stiffness in the turbine. It was clear now that colder temperatures usually resulted in better power; this could be caused by, for example, air density, cooling needed, or both.

To see how the forecast wind speed and turbine wind speed correlate, the wind speed mean was calculated from x and y vectors and compared to the wind speed in the turbine. Wind speed observations at Kauhava Airport were loaded from the Finnish Meteorological Institute to help. The correlation of airport observations and Met.no forecasts was around the 0,9 level, and they both correlated only around 0,65 with the turbine data. ChatGPT was used for help to calculate wind speed at a 130m height, and both FMI and Met.no were transformed; the result can be seen in Figure 16.

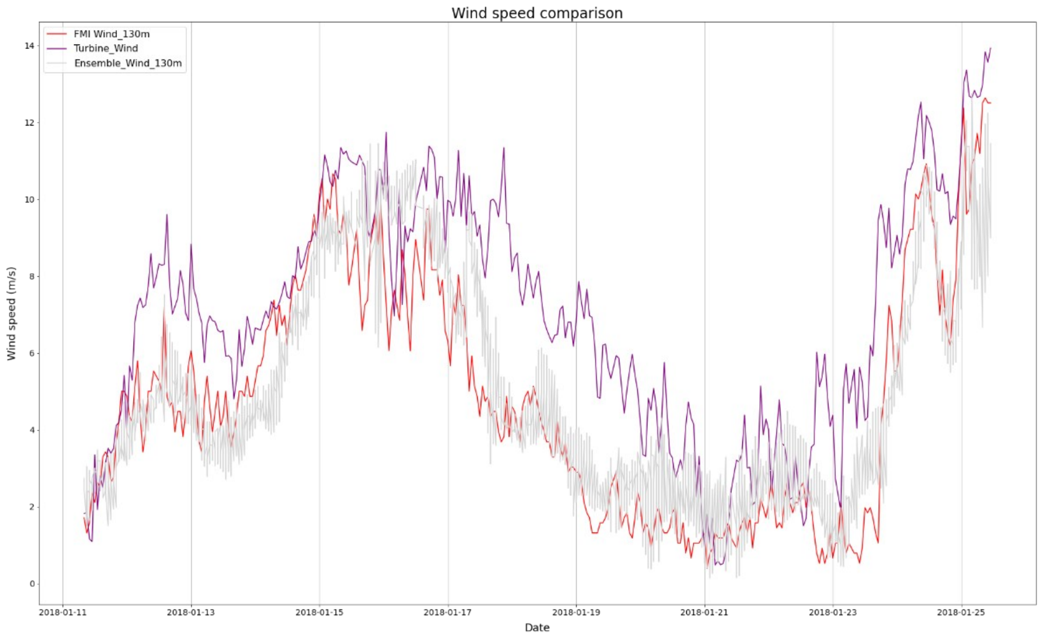

Figure 16. Observations versus forecast at 130m height.

The ensemble's thick lining is caused by the non-unique time index, but it shows the airport data following the ensembles better. A conversation with wind power representative Paavo Korpela from the Finnish Meteorological Institute revealed that the boundary layer causes the wind to react differently at different heights. This was already seen in early plots with projected winds at height compared to turbine observations. The work was already done, so nothing could be changed, but it shed light on the observed differences in the data. Wind at lower levels seemed to gain speed about as fast as at higher altitudes, but the speed dropped much faster at low levels.

One realization during the analysis was that there was no knowledge of the position of the turbine inside the data grid. The grid is 2,5 km × 2,5 km, and that would mean from corner to corner, the distance is 3,5 km. At 2,5 m/s, it takes 23 minutes to cross the grid, so if the turbine were located close to a corner, low winds from the direction of the opposite corner could technically correlate better with the power observation 20 minutes later.

To test this hypothesis, wind directions were divided into 8 sectors, time-shifted columns for multiple 10-minute shifts for each sector, and correlation for each. Initial trials looked promising, and more careful analysis showed some proof. (See figure 17.)

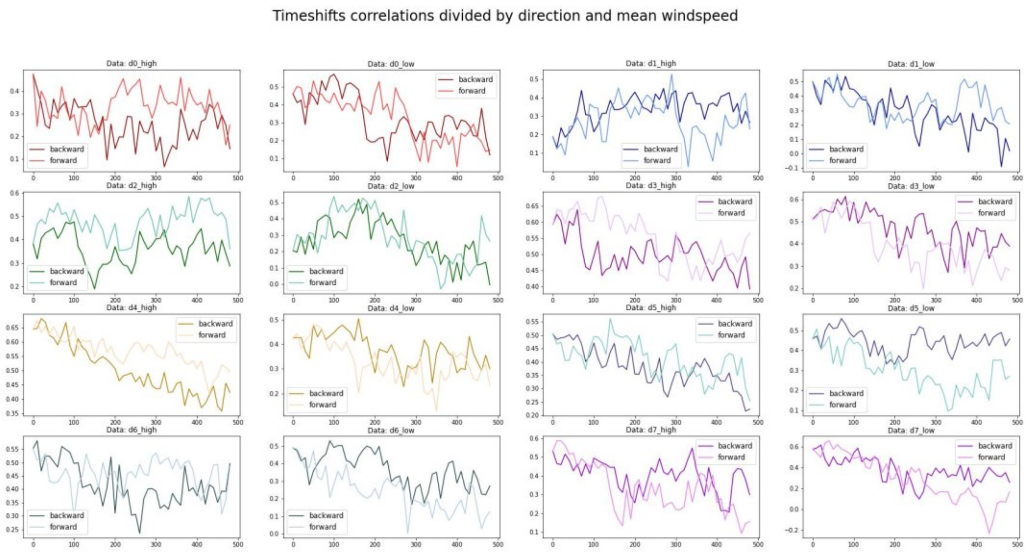

Figure 17. Wind speed correlations ±500 minutes, under and over mean wind speed.

Some increase in correlation can be seen in many of the low winds, but a big part of them are very low winds, which could cause more distortions in the correlations. Perhaps wind speeds should have been calculated separately, as in the earlier nacelle deviation analysis, but at that time, the gains for the work needed seemed too low. Maybe if the full data had been optimized, there would have been significant improvement, but at the time, it did not seem so. After all, the higher wind speeds should anyway be fast enough to pass the grid too quickly. The turbine could also be in the middle of the grid.

4.5 Challenges

4.5.1 Unsolved mysteries

Connection to the original data owner and commissioner Tuuliveikot Oy was missing throughout the whole project, and without that, no new information could be acquired. For example, it was unclear how the direction of the nacelle is managed or if the data is exactly as described. It was learned from the pamphlet of the turbine manufacturer that the turbine has an adjustable blade angle. Perhaps the earlier finding about the direction deviation could be explained by the blade angle, or the blade angle is just automatically used to cut higher-end power to the maximum, and the data is not recorded.

Contact with the Finnish Wind Power Association was attempted to find out some general knowledge about wind turbines, such as how much wind vane measurements can be trusted on top of the turbine. How much do the blades affect the wind speed and direction of the turbine data? A response after the first anniversary is expected to arrive soon.

The Norwegian Meteorological Institute (Met.no) offers other datasets than the currently used one, some with post-processing involved to better take into account statistical methods. Some of the post-processed datasets were started in 2018, and the number of variables was low for a low number of dates.

Several attempts were made to find a reason for bad predictions, in other words, find ways to improve. Neural networks are hidden in many ways, and when the actual physical formula was added to the neural network, the results did not improve at all. With the knowledge after the project was done, it is easier to see what was wrong, but while the process was ongoing, it would have been good to know more about how the neural network works in very deep detail.

One of the biggest revelations of the analysis was that the nacelle deviation does not work as expected. This was first seen in a 3-dimensional plot done with the Plotly library. To visualize the finding, it would be good to add that 3D plot to this document; for some reason, it turned out to be difficult.

Even when it is known that wind blows more at greater heights, defining the wind movements in the boundary layer is what people do as their full-day job, and understanding this came too late for this project. While it is possible to “simulate” the effect of height, it needs more information than just a data point to be able to transform the wind height with as simple data as at hand.

Ground roughness slows the wind faster near the ground, and the wind profile is modeled by professionals with greater knowledge than this project had.

4.5.2 Dataset creation

As the script for data loading was not created directly for this project, it had to be modified. Originally, it created a dataset that had unique dates and times, which made it easier to manage, combine with observations, and keep smaller. Having multiple 50+ hour windows of data that contained the same dates made it impossible to use simple data concatenations directly. In production, the input data will be one 36-hour dataset, and to match that, it was most likely better to use the same method for training.

The script did not consider that some of the files might be missing, causing the same data to be used for two different times. The error was not significant, but when multiple files were missing, it might have caused problems.

Data for the AIDA project was not for the same location either, so a whole year's worth of data had to be created from the start, which took almost a week with the script. Luckily, some data errors were already understood in the AIDA project, so they did not come as a surprise in this work.

Almost every iteration of the modeling process demanded a new way for the dataset, and every new way required a whole new dataset created from the loaded source data.

When interpolation was done, the data had wind speed and direction calculated. At that point, it was not realized that interpolation would cause the directions to be wrong when around 0 degrees. This method was later dropped, and new mistakes could be made. In the data engineering section, it is explained how the data was transformed, but one of the humorously ignorant moments was when the data that had been transformed to generic versions of the ensembles (min, max, mean, etc.) had a wind speed minimum greater than the maximum. It took a while to understand that the wind speeds were calculated from the minimum x/y vectors of the wind, which can be negative; only the “length” of the vector matters for the speed. When you have the largest wind of 10 meters per second in the “negative” direction, it is still 10 m/s, but it can be the minimum. So, the order of the calculations had to be switched.

4.5.3 Final testing dataset

The dataset for final testing had to be created through the same pipeline of transformations as the training data. Scaling was to be done with the scaler that was fitted with the training data, but after the transformation was done, column names had to be inserted back. This was done to make it easier to analyze the results and perhaps find reasons for the biggest errors. As the data from the end of the year was so bad, it could not be used for validation. To get the data for the dataset, all appearances of the dates were removed to ensure that there was no information left from the training set. This testing dataset was used as a validation dataset, mainly because a real test dataset would be done with new data, and we have no way to validate those results in the absence of new turbine data. This method enabled real-time decision-making for training.

4.6 Data engineering

Data manipulation and engineering consist of creating new data from, for example, domain knowledge. In the earlier half of the project, all possible data was added to improve precision. These variables included month, hour, and ‘hours from current time.’ The ‘hours from current time’ was made to show the model how many hours into the future the row would be and might have some value. Initially, the month and hour were added simply because it was possible, and usually, more data should only improve the predictions or have zero effect. There seemed to be predictive value in these variables, but the reason was still unknown. If there is wind of 10 m/s with air density being what it is, how would the month matter? As precision increased, it became evident that there was something behind it. Only in the end, through email exchanges with Paavo Korpela, did he reveal that the season and part of the day have a significant impact on how the wind profile works, and that was enough to explain what was seen already in the model. In the final version, these variables were installed as continuous values, so creating separate 0/1 values for all months, hours, and hours from the beginning might improve the model as they can be viewed as categorical variables.

Data cleaning done for frozen power was used as a label for the later iterations. The final dataset had four target variables: power, frozen, max power, and zero power. The AIDA project already indicated that the low parts might cause bigger problems for the error than the usefulness of the power output. This means if we could predict zero power somewhat accurately, simply setting it to

0 would reduce the error levels. It doesn't really matter if zero power is -50kW or 50kW. The late stage of the project showed that the used functions could not deliver negative values, and all power levels were increased by 100kW. Often, when the low or negative power could not be predicted accordingly, the error was very high. Full power was not focused on a lot, but it might be possible to also get better results if a separate model was done for each of these, and only the rows that go for power regression would be used for the error calculations. Frozen rows were only in the training data, so testing did not show accuracy for the prediction. Improving and transforming clustering these categories might improve the model precision; it might be worth investigating these possibilities later, but in the project, this was close to the end, and no more time was invested in this investigation.

Labels for freezing were originally made to clean the data from bad rows. A later iteration tested clustering by a couple of different algorithms like K-means, Agglomerative Clustering, and DBScan, but after initial testing, it seemed that so much work would go into this, and the original, more manual way was left in the data. In manual cleaning, three lines of power were added to cut all values below them as ‘stiff’ or frozen. (See figure 18.)

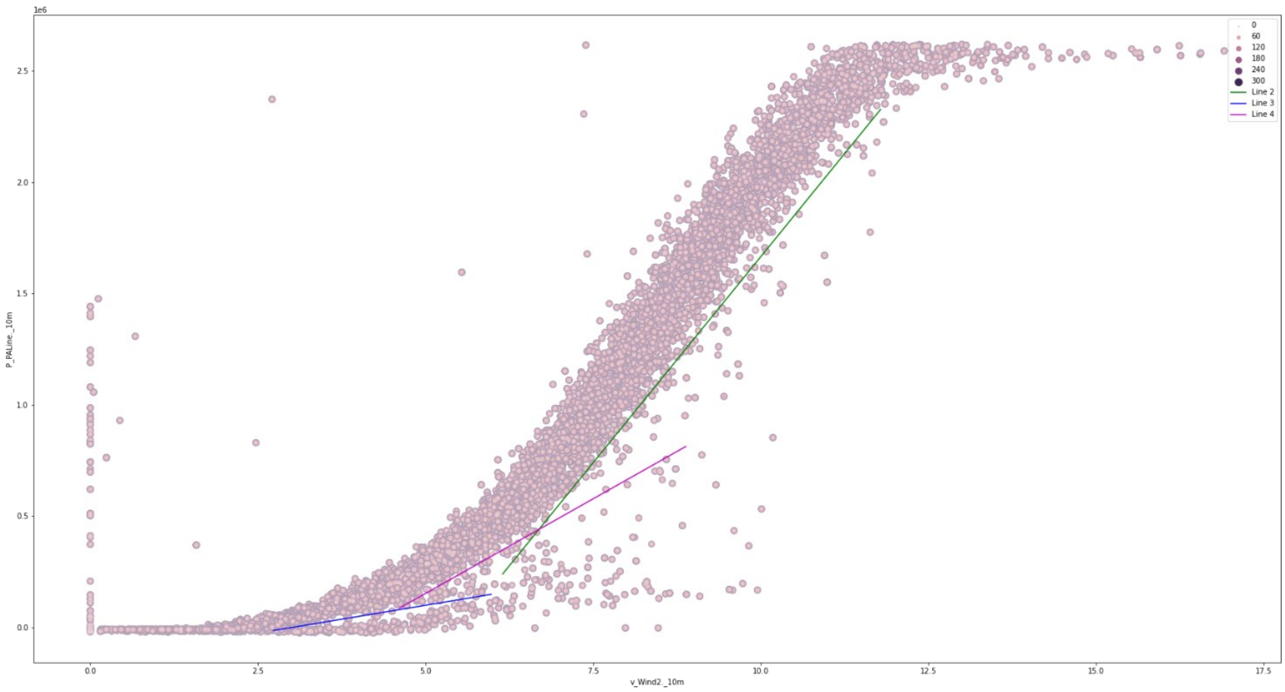

Figure 18. Data cleaning with lines.

The figure shows the simple data cleaning method, which works only for the clear outliers. However, if power is just down by a fraction, it would be inside the normal power range. Data points that are higher than the power curve mean that the turbine gives more power than the wind could give, indicating that the anemometer might be stiff. When doing a direct model for power output, the wind data is not used, and it is not a problem. The final version did not drop the frozen data but just the label as a target. Cleaned data that was tried for machine learning is in Figure 19.

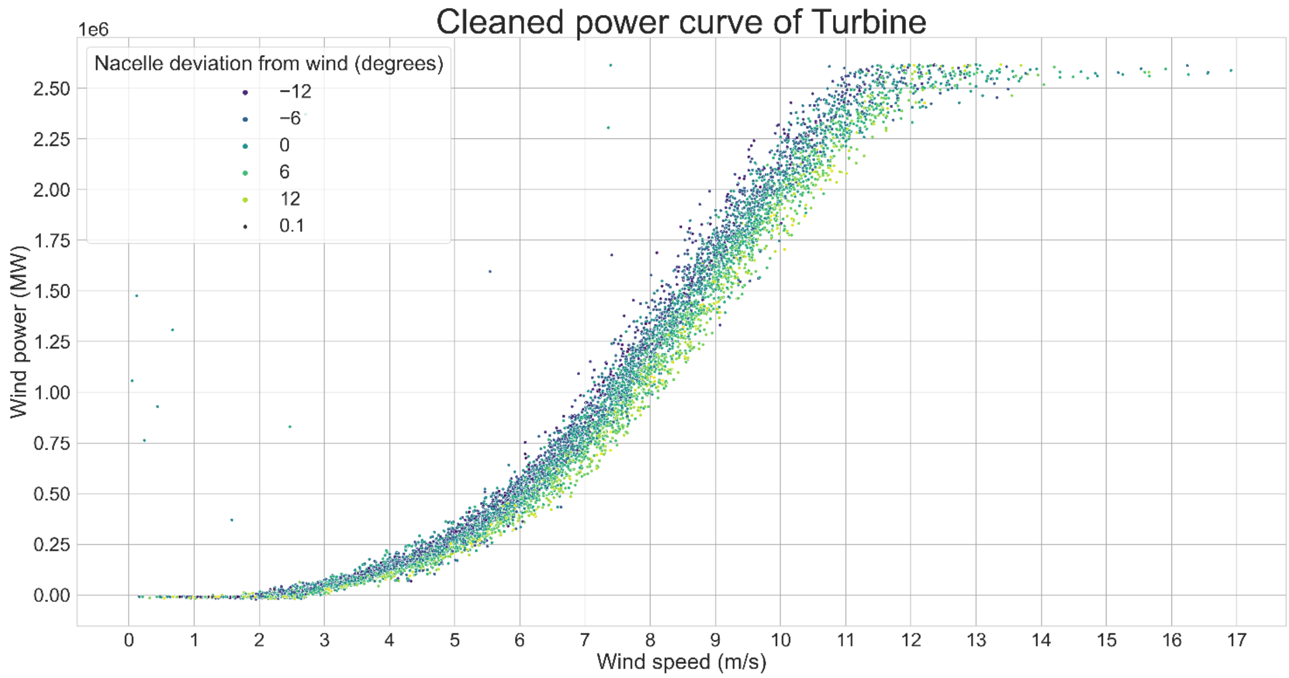

Figure 19. Cleaned power curve.

The machine learning path was dropped when the scale of the project was unveiled, but the cleaning work was not wasted as it could be used to label the data. To make the model work well with new data with 30 ensembles, the training data was transformed into different generic variables. Each row of 10 ensembles was transformed into min, max, mean, std, median, skewness, kurtosis, variance, and IQR (interquartile range). As many variables were created as one could think of to get the most information from the ensemble data. Percentile values and values closest to each percentile were also created. The AIDA project's results indicated that the error does not suffer extensively when mean, min, and max were used. Adding more was just to ensure the data is as good as possible. Final data had variables x, y, gust, humidity, temperature, pressure and density calculated for std, median, min, max, mean, variance, skewness, kurtosis, IQR range and 10, 25, 75 and 90 percentiles and values closest to those percentiles and deviation angle in radians.

4.7 Modelling

4.7.1 Model architectures

Along with many iterations in the project, there were tens of different models. Most of the time, parameters were adjusted by trial and error. In the beginning, the results improved as long as the training continued. This raised eyebrows since the validation score also improved almost as quickly as the training score. The project was under the impression that overfitting should be avoided, and a lot of research has been done about it. Still, in this project, it seemed that overfitting was beneficial. That model was done with a dataset of unique time indexes and one target variable (power).

The next iteration was done by separating all ensembles into their own rows, creating 10 times the data rows that were less wide. An ensemble number category was added to the data to distinguish between different ensemble members. The idea behind this was to create a model for each ensemble and compare them.

Different methods were added to the model to create different paths for different data; categoricals followed a simpler path. Some iterations had static variables, like the area of the propeller, added to act as a multiplier, but the expectation was that the neural network should not need this kind of data. Activation layers were tested as much as possible, and some custom loss functions were created to calculate the data according to the physical formula.

The last iteration focused on simplifying the model to speed up the testing process. The latest model architecture is in Figure 20.

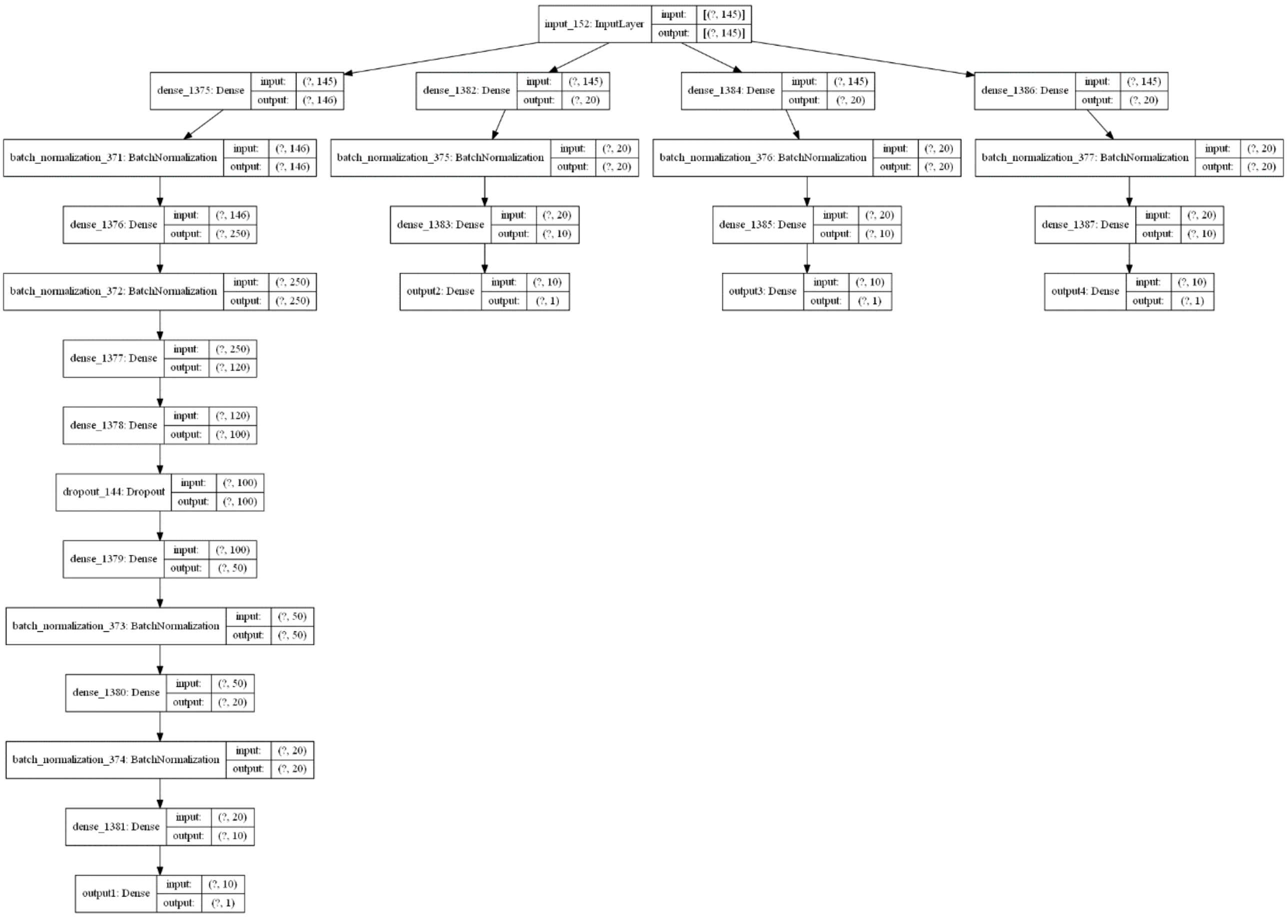

Figure 20. Final model architecture.

Predicting the different groups helped mostly at the lower end of the power output, which was set programmatically to mean value of the negative powers.

4.7.2 Training helpers

During the early stages, with model training taking extremely long, it was strange that even when training the model for an entire day, the model did not seem to use model from the best epoch. Then it was found that the EarlyStopping callback function has a 'restore_best_weights' option. This function would stop the training when enough time had passed since the last improvement in score. The same investigation found another callback to try to avoid situations where the model could not find a way to better predictions due to the learning rate causing the model to jump over the path that would lead to better results. The ReduceLROnPlateau function enables the learning rate to get lower if no progress has been made. The LearningRateScheduler could also drop the learning rate, but it uses a set rate for the decrease, regardless of whether the score still increases.

4.8 Predictions with Machine learning

As this part was predicting wind velocity at 130 meters, the training target data had to be a cleaned version of wind observations. Zero wind caused some issues too, because it was difficult to know from observations that the wind speed was faulty until analysis with the power was done. After cleaning the dataset, the search for the best algorithm could start. Machine learning was meant to be done in two parts: the first part was using regression of nacelle-level wind from predictions. For this, multiple different algorithms were tried. LinearSVR, SVR, Linear Regression, XGBoost, RandomForestRegressor, DecisionTreeRegressor, GradientBoostingRegressor, and SGDRegressor were all tried as the work progressed. Most work in this approach was put into finding the best possible hyperparameters for each of the regressors. Optimizing hyperparameters was done with grid search, but at the time, it felt that the progress was very slow for half of the project. For this reason, this approach was dropped entirely. A lot of work would have been needed to predict wind speed and then again for the power.

With the knowledge gathered throughout the project, it would be a very plausible way to get good results. Getting more value than the neural network would still need the real model for wind at 130 meters. In this approach, the work would have had the same problems, with no way to improve the predictions without time series.

4.9 Deep learning and Neural Networks

Work with neural networks went deeper into all aspects of the modeling. The data model was planned to have readiness to be implemented into a production system. A lot of thought went into figuring out a model that would imitate the real world the most.

The power curve resembles the sigmoid function, so one iteration had a sigmoid activation layer in the model. This was a desperate attempt to find ways to improve the results. A test was made to see if an LSTM could predict total power output for 36 hours better than this one-hour prediction could, at least on average. This test needed a new dataset that had every day as a dataframe of 36 hours, and the target variable was the sum of 36 hours. The same dataset could be used for a convolutional neural network (CNN).

The final model had three categories to predict and one continuous value. Some testing was done to figure out what kind of weights each of the predicted values had in the model. In TensorFlow 2.1 and older, it was possible to adjust weights manually, but for some reason, the option was disabled. See the appendix for the code.

4.10 Mistakes

In the early model, the interpolation of training data to 10-minute intervals was supposed to be a way to increase the amount of data. Usually, it would be helpful to have as much data as possible. However, this interpolation did not help with missing yearly data, which would have brought more frozen data points to aid in predictions concerning freezing. The way this interpolation was used caused significant data leakage. The only guess about the nature of the data contamination was that when data gets imputed from point X to point Y, with averages, the row has some information about the origin point and destination. When these data rows are then shuffled into training data and validation data, the validation data has information about the training data and vice versa. This caused very high overfitting and seemingly endless predictive power. When predictions (in validation) with a mean absolute error of under 100kW felt acceptable, and at the same time suspicious, some testing had to be done to see how the totally new data would be predicted.

Testing data predictions had a very high mean absolute error, and it seemed that even though model training looked extremely good, the model did not learn anything. The model would be unusable with over 600kW mean absolute errors, and the reason for the high error had to be found. A scatterplot made with the predicted output and observations looked totally random. There was some correlation, but it was nowhere near the validation score. The data had 54 hours, and the predictions were binned into 9 groups to distinguish how many hours into the future the prediction was. (See figure 21.)

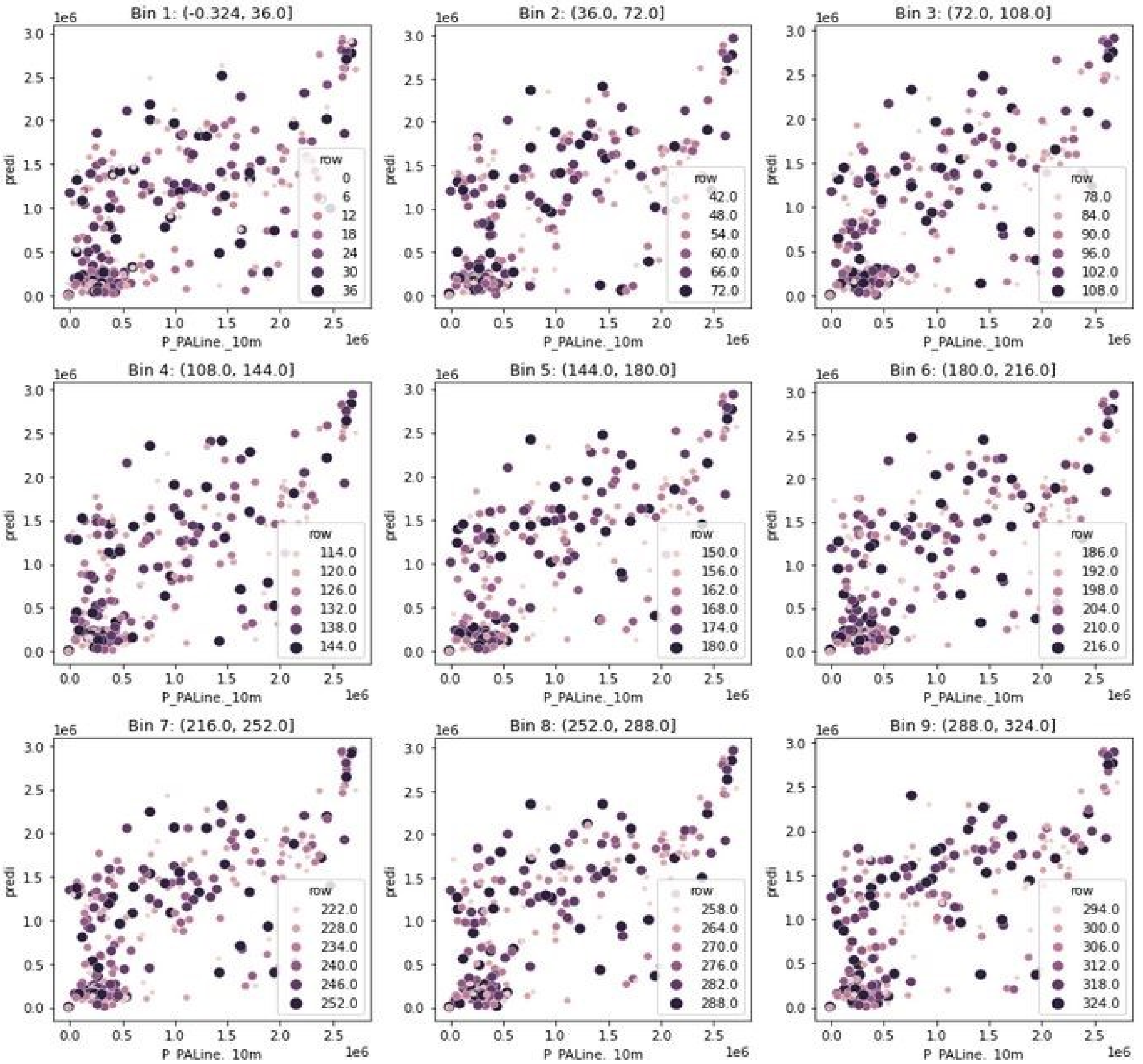

Figure 21. Predictions versus observations divided into bins of elapsed hours, colored by row number.

These figures showed no correlation with how many hours had passed. The row values are the 10-minute intervals. Later, when interpolated data was removed, the same row number was still in the data, but hour 1 would be row 6.

After this image was drawn, it was certain there was an error in logic. When the reason for the error was realized, the interpolation was dropped. Perhaps it could have stayed, but some adjustments to the validation and testing should have been done. Early stopping never activated when it was monitoring the validation error, which always went down, and at that point, the model was extremely overfitting, and test data predictions were poor.

5 Results

5.1 Results

When the error with interpolation was fixed and work was started again with new data, the results were bad, almost as bad as the overfitted testing. After that, a lot of thought was put into figuring out various ways to improve the precision. Many of the ways were not possible or did nothing to help. At that point, it was important to forget the levels of error from before and settle with the new levels; after all, they were results that could be trusted. Earlier 100kW levels were achieved by training the model with validation data and would be almost the same as using observations of wind speed to predict the power.

In the final iteration of the project, the emphasis on precision polish was diminished as steady training became more important along with a shorter iteration cycle. After a lot of work was done with variables and data models, the training usually needed only 10 epochs to find the best score. The mean absolute error (MAE) for all predictions was around 300kW. For predictions that were smaller than the observed power, the MAE was at 270kW, while over-predictions had over 430kW. A simple comparison with linear regression using the same data gave an MAE of 470kW, which also shows that there was some learning in the model.

The learning rate was adjusted while training was done, and in Figure 22, the model training process can be seen along with validation data, which was the same data that was used for testing the precision after calibration.

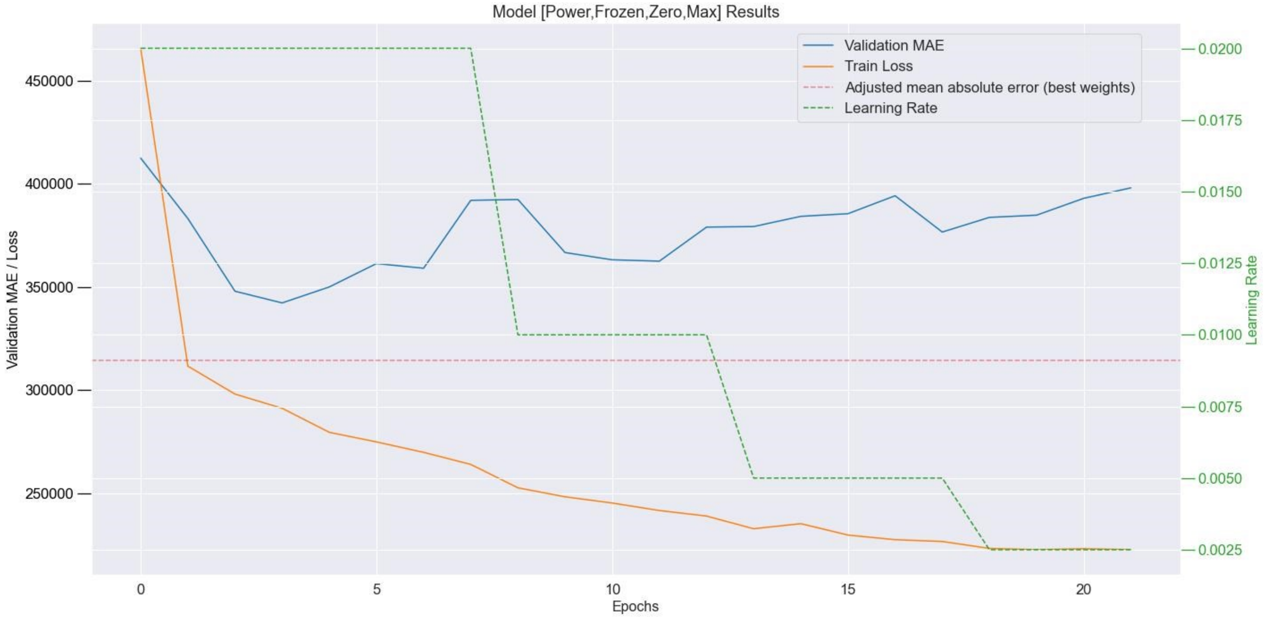

Figure 22. Training errors versus learning rate.

Predictions done with this data resembles a regression but there are big errors all around. Figure 23 shows the power observations on X-axis and predictions on Y-axis. A black line was later added to mark the perfect regression.

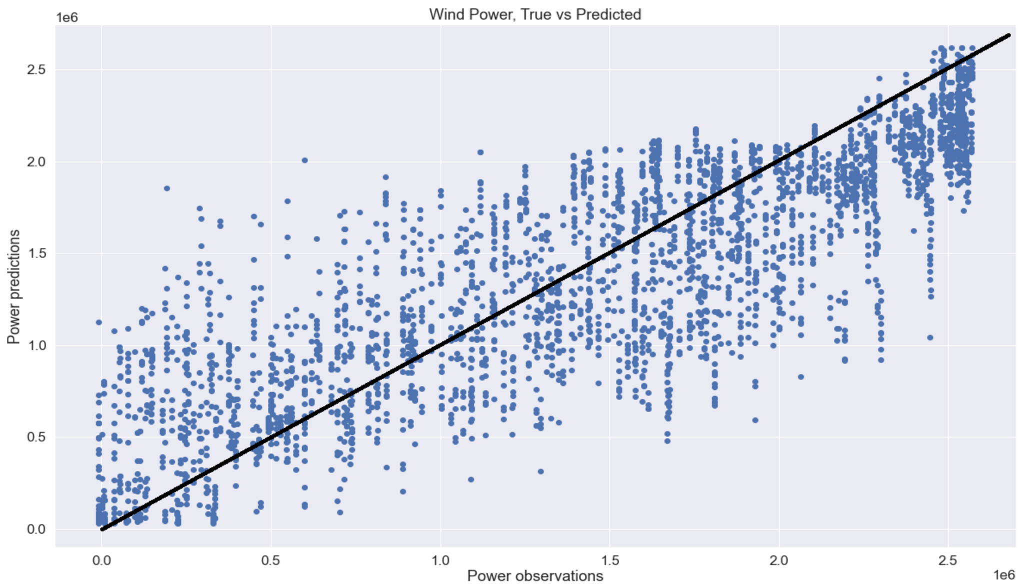

Figure 23. Predicted power versus observed turbine power.

Output2 of the model was predicting freezing and as there was no frozen labels in the summer data used for testing, the predictions were all true. Output 3 and 4 had occurrences in the test data, as the output3 was so called zero power, and output4 was maximum power. Confusion matrixes for these outputs can be seen in Figure 24.

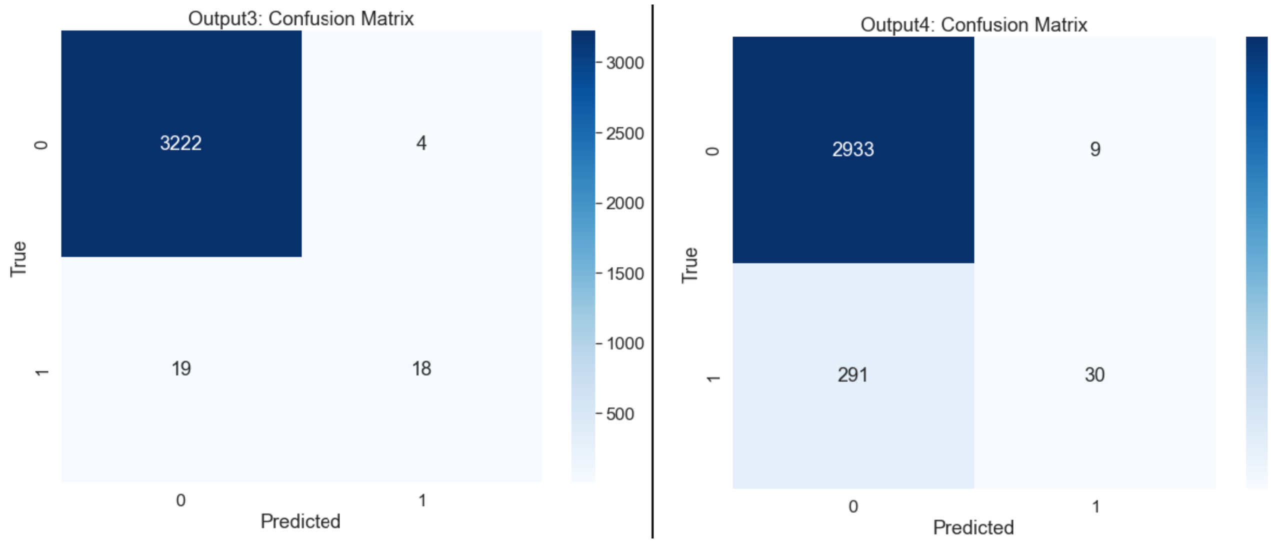

Figure 24. Outputs 3 and 4 predicted.

5.2 Journey to last results

After a lot of preparation for the last iteration, the first prediction had a validation MAE of 550kW. This test was done with December data, and it is understandable that the results were not the best. Test data had to be reasonable and correct. Significant improvement was gained when more variables were added to the model. First iterations only had wind in different formats and temperature. Adding components of air density reduced the error level significantly, and including month and hour improved the error level by over 10%.

6 Discussion

6.1 Findings

When data analysis started, it was not clear what the data could show and what the findings would be. Ready tools for exploratory data analysis are available, but when they were tried, they seemed so basic that they were of no use for real analysis. The biggest findings in analysis came as accidental interruptions. When the model is supposed to predict a value dependent on another value formed with an unknown model (nacelle direction), the only way to fix it would have been to correct the data to the maximum the turbine could get. That would have required a nacelle deviation that has the maximum power, which was supposed to be directly towards the wind. Reasons for the nacelle issue can be anything, from turbine blades interacting with wind speed and direction measurements to actual output increase. Errors in logic and code can happen as well, but as turbine optimization was not within the scope of the project, the investigation should be done by more specialized individuals.

Nevertheless, if the nacelle were pointing 10-20 degrees to the negative direction according to the turbine's own vane, the power output should increase. If this is a phenomenon that can be observed in turbines worldwide, the increase in power generation from wind energy could be significant, perhaps even 10%. According to ChatGPT, in 2023, global wind power generation was about 2300 TWh, and even a 5% increase would be 115 TWh. That is about 8 times the Finnish wind power production of that year.

Before the project, a neural network was seen as an artificial intelligence that would solve any problem if more data was thrown into the black box. This was proven to be a faulty assumption. The model is only as good as the data that is put in, which was proven true. There are many ways data leakage can happen, and very often, after too much work with an idea, the results jumped, and the reason was the accidental implementation of wrong data.

The learning curve of the project was long and steady. Many ways to create a model were tried, and it was disappointing to realize that the phenomenon is so simple to model that it works almost as well with any kind of model. The model made with only ensemble 0 wind speed might be very close to what this work could accomplish, just because of how important the wind profile projection for higher altitude is. The only real prediction in this model would be the air density; with corrected wind speed, it would be possible to calculate the power by hand. It would be interesting to see how a small amount of data from this one year would be enough to provide as good results.

The training data had temperatures of 212 Kelvins and over 150 m/s winds, and it is still a question if and how much the data should be corrected. In this case, the obviously broken forecasts are in the most unreliable ensemble member, and can happen in the future also, and should not have a lot of weight, but in general precision seemed to drop whenever data was removed.

After a lot of work, it seemed that nothing other than wind speed transformation would improve the performance to new levels.

At this point, the discussion with the FMI representative forced a change in scoring method to R-squared so the results could be compared to their score. R-squared or the coefficient of determination measures the variance in the target variable that was predicted. Paavo Korpela from FMI said their model had an R-squared of around 0.8. This model was usually at around 0.75, ±0.3. This was a sign that the work was complete, since even though the mean absolute error of 300kW in a 2.5MW turbine is not extremely good, it is still creating value for the owner as the errors can easily exceed over 500kW when using simpler methods. Using only common weather forecasts that use wind speed integers, and with air density missing, perhaps even an error of 1MW could be seen. Results this close to the ready-made model from FMI showed that something was done right.

Without knowledge of the business aspect of this turbine, it is difficult to predict how much value this model would bring in the end. One negative side of the results is that the model has a smaller error in the values it predicts too low, and the error is bigger when the prediction overshoots. The good side is that out of 2195 rows, the model predicted 1630 rows lower than observation and only 565 rows higher. This is a good sign as it should be better to promise too little energy than be forced to buy reserves.

Creation of different data models was slow work, and often the dataset had to be created from original data loads, which slowed the process even more. In a simple model like this, the amount of work did not pay off. Wind power is so simple that the model did not need many epochs and usually different model architectures did not help either. The more complicated the data model and model architecture is, the easier it is to make fatal errors with the logic.

The results did not try to find reasons for the errors; this was tried before the project was reset. With the knowledge now, it is most likely that the error levels are coming from the wind profile and boundary layer dynamics. The model was not trained with a single-dimensional data index either. This option might have revealed if the results would be better if the training data did not have data so many hours into the future. The guess is that this would, in fact, reduce the precision.

6.2 Future

One important thing to note about the future is that in 2023, the electricity market started to use a precision of 15 minutes in power sales. Taking advantage of this model might require going back to the interpolated data for input data. The latest model was done with data of 36 hours. This should be reconsidered, since a 36-hour slice in the wrong time zone would give only a 34-hour forecast for Finland. The last training was done with the average of two time points because the forecast data is point-in-time. This means another hour is lost. One simple trial for improvement would be to take the wind mean of the last 6 hours as columns for the rows. This way, the model would get knowledge of the forecasted wind speed for 6 hours prior to the current time. Perhaps the forecast of the next 6 hours could be taken as well; the data is delivered and available 10+ hours after the 36+3 hours.

The new 30 ensemble forecast data also has a bigger grid, which means that the same grid number does not have the same locations. This does not matter in the load script, as the coordinates are transformed to grid positions. Still, this is a detail that was found quite late, and for a moment, it caused mild panic.

With more data, it would perhaps be possible to create a better model for freezing detection, but with only one year of data with a relatively small number of occurrences, the results were not great. For this kind of task, outlier detection would be a better approach if the data is low.

To really improve this model, the first task would be to transform the wind speed to better resemble the wind at the height of the turbine. Knowledge acquired from the project reveals that when the wind has been transformed to height, results might be quite good even with only wind speed used. Other variables are air density and the nacelle direction. A mean value could be used for air density, and perhaps greater minds could figure out a way to go around the nacelle direction problem. Verifying the nacelle deviation observations with publicly available data would be a good and most likely fast task.

Multiple ways to try to gain better predictions were planned, but when results got so close to the FMI score, they were put on hold until next time. One way was to pick categories for the results, group the power in 300kW bins, and try to predict the group. This, on average, would be half of the mean absolute error, but obviously, the model would need very high accuracy for that to be worth it. Post-processed data might be somehow compatible with the model, or the model could be made to accept it. This project did not need it, since there was no new turbine data to know how the model would work.

A small gain might come from investigating power consumption when the nacelle is turning, and how it affects the power output. If the effect is large enough, adding a flag for turning to the data might make the model work better.

To create working predictions for Tuuliveikot Oy to use for their business, the model must be put online. An easy way to do this is to download the new data and insert that data into the model every 6 hours. This would hand out the predicted values, and they could then be delivered, for example, to a web page. Data load and transformations would need to be automated in the pipeline. Loss calculation for freezing or other malfunctions might be useful in the future when projected power becomes more accurate. The model could push out expected power, and when a certain threshold has been crossed, there is a malfunction.

6.3 Conclusion

The objectives for the project were to investigate if the data delivered by the Norwegian Meteorological Institute can be used to predict wind power accurately enough. The original definition for "good enough" was “as good as possible.” While it was good to have no pressure, it also made it possible to continue polishing the results forever. The accuracy of the application made by the Finnish Meteorological Institute helped here a lot. A small disappointment was experienced when this other competing application was discovered. It is still quite safe to say that this objective was successful. Weather forecasts are done from the same data used here, but with a specialist's touch, and the data gained for this project was well suitable for the need. Predictions having an average error of 300kW are not extremely good, but with possible ways to improve already found, the model could be better than FMI's one.

Another objective was to find ways to create value by analyzing the data. The analysis done here found possible ways to increase the output, and they would just need a peer review to see if this can be verified for other turbines as well. So that objective was completed, and the future will hopefully show how valuable these findings were.

This project was, for some time, done simultaneously with the AIDA project, and that was a big mistake. It was very easy to mix up the work, starting to analyze AIDA results compared to wind turbines data, for example. The AIDA project had the same issue with problem description and desired outcome—there was none.

Another flaw with this project was the wide knowledge needed. At first, the problem and approach seemed very simple, as it should have been. Unfortunately, the physics part, which was not meant to be, at least, this big a part of the thesis, was more complicated than expected. Having so little domain knowledge about the physical aspect created big difficulties in understanding the reasons for the challenges. Bad results were blamed on the model, data model, and model architecture, while the problem as a machine learning task was simple, but wind physics created a problem that could not be solved with only this format. A lot of time would have been saved if this was understood earlier. While there was plenty of technical advice, the physical side received help only at the very end.

As it was valid logic not to have time series forecasting for the power, because the hours are independent and wind time series work is already in the data model that creates the used data, it might have had better results immediately with time series because of how the wind works in the bigger picture. This would work perfectly if the turbine was at a height of 10 meters.

During this project, I had quite a good path of learning, perhaps for the wrong reasons, as nothing I did made the results notably better. Creating my own loss functions and custom layers taught me a lot. I had to dive a bit more into the mathematical side of the models to understand some of the reasoning that goes on in the model. Also, realizing that the neural network is just a tool that does pretty much what more traditional algorithms do, especially when the task is not understood exactly. The project also taught me the importance of definitions and design of the project. Composing this document was extremely hard work when, during the work, the format of the document was not clearly in my mind. Creating a document from random idea bullet points was time-consuming and nerve-wracking. First, the bullet points were organized into logical groups where ideas or findings belonged, and then I made a document format just to realize that, because they were not compatible, all ideas and findings were mixed with stories that belonged to different parts of the document, text, and timewise.

There were times when I realized that I could continue to investigate this data for years just to understand everything about it, and there were times when I almost decided to deliver the old version that had totally nonsensical results and many ideas were already debunked.

All in all, the journey of this project was difficult in many ways, but it taught me a lot about project management, or at least the value of it, and determination. I had to get results that made any sense.

References

Examples. N.d. Norwegian Meteorological Institute. Accessed on 22 May 2024. Retrieved from https://github.com/metno/NWPdocs/wiki/Examples

Heikkola, Erkki. 2016. Tuulivoimakohteen melu-, välkevarjostus- ja näkyvyysmallinnukset [Wind power site noise, shadow flicker, and visibility modeling]. Accessed on 31 October 2022. Retrieved from https://www.kauhava.fi/files/14382/Liite_10_Melu-_ja_varjostusmallinnusten_selvitys.pdf

Hutchinson, Mark. Zhao,Feng. 2023. Accessed on 7 December 2023. Retrieved from https://gwec.net/wp-content/uploads/2023/04/GWEC-2023_interactive.pdf

Kalmikov, Alexander. 2017. Massachusetts Institute of Technology. Wind Power Fundamentals. Accessed on 31 October 2022. Retrieved from http://web.mit.edu/wepa/WindPowerFundamentals.A.Kalmikov.2017.pdf

Korpela, Paavo. 2024. Meteorologist, Account Manager, Renewable energy. Finnish Meteorological Institute. E-mail exchange. 18 March 2024.

Kujala, Ari-Pekka. 2024. Met.no sääennustedatan hankkiminen ja analysointi [Acquisition and analysis of Met.no weather forecast data]. Accessed on 26 May 2024. Retrieved from https://gitlab.labranet.jamk.fi/tieto-tuottamaan/metno/metno-about

L100 2.5MW. 2018. Brochure from Lagerwey. Lagerwey. Accessed on 31 October 2022. Retrieved from https://www.lagerweywind.nl/wp-content/uploads/2018/02/7105-LW-Leaflet-L100-ENG_WEB.pdf

OX2 osti tuulivoimahankkeen Kauhavalta [OX2 purchased the wind power project in Kauhava]. 2022. STT Viestintäpalvelut Oy. Accessed on 31 October 2022. Retrieved from https://www.sttinfo.fi/tiedote/ox2-osti-tuulivoimahankkeen-kauhavalta?publisherId=69817516&releaseId=699347204

Tiedot hankkeesta [Project information]. N.d. OX2. Accessed on 6 May 2024. Retrieved from https://www.ox2.com/fi/suomi/hankkeet/salo-ylikoski/

Tuulivoima Suomessa 2022 [Wind power in 2022 in Finland]. 2023. Finnish Wind Power Association. Accessed on 7 December 2023. Retrieved from https://tuulivoimayhdistys.fi/media/tuulivoima_vuositilastot_2022-1.pdf

Wind power in 2023 in Finland. 2024. Finnish Wind Power Association. Accessed on 22 May 2024. Retrieved from https://tuulivoimayhdistys.fi/media/wind-power-in-finland-2023.pdf

Wind power production increased by 25% in 2023 – domestic wind power increases electricity availability. 2024. Finnish Wind Power Association. Accessed on 22 May 2024. Retrieved from https://tuulivoimayhdistys.fi/en/ajankohtaista/press-releases/wind-power-production-increased-by-25-in-2023-domestic-wind-power-increases-electricity-availability

Wind Speed Calculator. 2003. Danish Wind Industry Association. Accessed on 31 October 2022. Retrieved from http://www.drømstørre.dk/wp-content/wind/miller/windpower%20web/en/tour/wres/calculat.htm

Appendices

Appendix 1. Code for final model and training

```
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Dense, Concatenate, Lambda,Reshape,Flatten,BatchNormaliza-
tion
from tensorflow.keras import regularizers
import tensorflow.keras.backend as K
from sklearn.utils.class_weight import compute_class_weight

def root_mean_squared_error(y_true, y_pred):
    return K.sqrt(K.mean(K.square(y_pred - y_true)))

pd.set_option('display.max_rows', 20)

modelGUGU = 0
# Define input layer
df_1_2 = df_m1.copy()
df_1v_2 = df_m1v.copy()
df_1_2['month'] = df_1_2.index.month
df_1_2['hour'] = df_1_2.index.hour
df_1v_2['month'] = df_1v_2.index.month
df_1v_2['hour'] = df_1v_2.index.hour
inputs = Input(shape=(df_1_2.shape[1],))

x = Dense(df_1_2.shape[1]+1,activation='relu')(inputs)
x = BatchNormalization()(x)
x = Dense(250,activation='relu', kernel_regularizer=regularizers.l1_l2(l1=0.04, l2=0.01))(x)
x = BatchNormalization()(x)
x = Dense(120,activation=LeakyReLU(alpha=0.4))(x)
x = Dense(100,activation='relu')(x)

x = Dropout(0.50)(x)
x = Dense(50,activation='relu', kernel_regularizer=regularizers.l1_l2(l1=0.03, l2=0.03))(x)
x = BatchNormalization()(x)
x = Dense(20,activation=LeakyReLU(alpha=0.4))(x)
x = BatchNormalization()(x)
main_branch = Dense(10, activation='relu')(x)

y = Dense(20,activation='relu')(inputs)
y = BatchNormalization()(y)
catekorit1 = Dense(10,activation='relu')(y)

y = Dense(20,activation='relu')(inputs)
y = BatchNormalization()(y)
catekorit2 = Dense(10,activation='relu')(y)

y = Dense(20,activation='relu')(inputs)
y = BatchNormalization()(y)
catekorit3 = Dense(10,activation='relu')(y)
```

```
# Define output layers
output1 = Dense(1, activation='linear', name='output1')(main_branch)  # Linear regression output
output2 = Dense(1, activation='sigmoid', name='output2')(catekorit1)  # Binary classification output
output3 = Dense(1, activation='sigmoid', name='output3')(catekorit2)  # Binary classification output
output4 = Dense(1, activation='sigmoid', name='output4')(catekorit3)  # Binary classification output


# Define model with multiple outputs
modelGUGU = Model(inputs=inputs, outputs=[output1, output2, output3, output4])

Y1_train = target_med.iloc[:, 0] +100000 # First column
Y2_train = target_med.iloc[:, 1]  # Second column
Y3_train = target_med.iloc[:, 2]  # Third column
Y4_train = target_med.iloc[:, 3]  # Fourth column

Y1_val= target_val.iloc[:, 0] +100000
Y2_val= target_val.iloc[:, 1]
Y3_val= target_val.iloc[:, 2]
Y4_val= target_val.iloc[:, 3]

optimizer = tf.keras.optimizers.Adam(learning_rate=0.02)
# Compile the model
modelGUGU.compile(optimizer=optimizer, loss={'output1': 'mse', 'output2': 'binary_crossentropy', 'out-
put3': 'binary_crossentropy', 'output4': 'binary_crossentropy'}, #, 'output4': 'binary_crossentropy'
             metrics={'output1':'mae','output2': 'accuracy', 'output3': 'accuracy', 'output4': 'accuracy'})#

# Train the model
hist = modelGUGU.fit(df_1_2, {'output1': Y1_train, 'output2': Y2_train, 'output3': Y3_train, 'output4':
Y4_train}, epochs=2000, batch_size=42,
           validation_data=(df_1v_2, {'output1': Y1_val, 'output2': Y2_val, 'output3': Y3_val, 'output4':
Y4_val})
          ,callbacks=[

   tf.keras.callbacks.ReduceLROnPlateau(
   monitor='val_output1_mae',
      patience=4,
      factor=0.5,
      min_lr=0.00001,
      cooldown=2 #ennen 10
   ),
   tf.keras.callbacks.EarlyStopping(monitor='val_output1_mae', patience=18,restore_best_weights=True
)

])
```