CHALMERS
UNIVERSITY OF TECHNOLOGY

Machine Learning for Wind Power Prediction

A Comparative Analysis of Traditional Machine Learning Models and Graph Neural Network for Wind Power Prediction and Forecasting in Wind Farms

Master's thesis in Mechanics and Maritime Sciences

Emma Antonsson
Lisa Vind

DEPARTMENT OF Mechanics and Maritime Sciences

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Machine Learning for Wind Power Prediction

A Comparative Analysis of Traditional Machine Learning Models and Graph Neural Network for Wind Power Prediction and Forecasting in Wind Farms

Emma Antonsson
Lisa Vind

CHALMERS
UNIVERSITY OF TECHNOLOGY

Division of Fluid Dynamics
Chalmers University of Technology
Gothenburg, Sweden 2025

Machine Learning for Wind Power Prediction
A Comparative Analysis of Traditional Machine Learning Models and Graph Neural Network for Wind Power Prediction and Forecasting in Wind Farms
EMMA ANTONSSON, LISA VIND

Supervisor: Hamidreza Abedi, RISE
Examiner: Lars Davidson, Mechanics and Maritime Sciences

Master's Thesis 2025
Division of Fluid Dynamics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Image of wind farm. Photo by Luca Bravo on Unsplash.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Machine Learning for Wind Power Prediction
A Comparative Analysis of Traditional Machine Learning Models and Graph Neural Network for Wind Power Prediction and Forecasting in Wind Farms
EMMA ANTONSSON, LISA VIND
Division of Fluid Dynamics
Chalmers University of Technology

Abstract

This thesis presents a comparative study of machine learning (ML) models and the deep learning (DL) model graph neural network (GNN) for wind power prediction and short- to medium-term forecasting in wind farms. Using high-resolution SCADA data from a 16-turbine onshore wind farm in Sweden, along with re-analysis and forecast weather datasets, various models including Random Forest (RF), eXtreme Gradient Boosting (XGBoost), k-nearest neighbours (kNN), Multi-Layer Perceptron (MLP), and GNN were trained and evaluated. Two baseline approaches, the farm's theoretical power curve and FLORIS wake model, were used as references. Results show that ML models outperform baseline models in predicting wind power output, with GNN achieving the best overall performance, although all ML models perform similarly. The ability of the models to generalize from wind power prediction to forecasting is however limited. The findings indicate that re-analysis data with low spatial resolution fails to adequately capture local weather conditions necessary for accurate power prediction. The study also investigates the effects of input feature selection, temporal resolution, and multi-task learning on model performance. Furthermore, it identifies challenges related to input data quality, particularly in the estimation of global wind conditions from SCADA-based measurements. These results underscore the potential of ML methods for wind power applications and highlight the critical importance of accurately representing global weather data, as well as accounting for discrepancies between training data and forecast data.

Keywords: wind power prediction; machine learning; graph neural network; SCADA data; wake effects; power forecasting; multi-task learning; north calibration

Acknowledgements

Major thank you to our supervisor Dr. Hamidreza Abedi for your support and guidance in this master thesis. Throughout the process your door has always been open to us and we are grateful for your willingness to answer any questions, explain and explore ideas. Big thank you to our other supervisors at RISE, Dr. Maria Bånkestad, Dr. Leon Sütfeld and Dr. Aleksis Pirinen. We truly appreciate the time and insight you shared with us during our weekly meetings. Your guidance, ideas, and expertise have taught us a great deal, and we're grateful for your support throughout this journey. We also want to give a big thank you to our other colleagues at RISE, Dr. Saptarshi Sarkar, Dr. Rémi Corniglion, Dr. Johan Arnqvist. Your knowledge in wind power and weather data, along with the insightful discussions, the ideas and work you have shared, have been very valuable to us. You have been an incredible support, and we truly appreciate it.
Thank you to our examiner professor Lars Davidson. For overseeing the project and its progress. We also want to thank Sarah Nilsson at SR Energy for taking the time to answer any questions about the wind farm and data used in this project. Thank you to the NAISS supercluster and the computational resources we have been provided. They have enabled us to train our machine learning models and this project would not have been possible without them. Additionally, we're grateful to the Chalmers Writing Centre for their valuable support and guidance throughout the thesis writing process.
Finally, to our friends and family, thank you for supporting us through out our time at Chalmers.

Emma Antonsson, Lisa Vind, Gothenburg, June 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

AEP	Annual energy production
AI	Artificial Intelligence
API	Application Program Interface
AR	Autoregressive
ARMA	Autoregressive Moving Average
DL	Deep Learning
ECMWF	European Centre for Medium-Range Weather Forecast
FCR-D	Frequency Containment Reserve Disturbance
FLASC	FLORIS-based Analysis for SCADA data
FLORIS	FLOw Redirection and Induction in Steady State
GNN	Graph Neural Network
IFS	Integrated Forecasting System
kNN	k-Nearest Neighbors
LSTM	Long-Short Term Memory Network
MAE	Mean Absolute Error
ML	Machine Learning
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
NREL	National Renewable Energy Laboratory
NTM	Normal Turbulence Model
NWP	Numerical Weather Prediction
ReLU	Rectified Linear Unit
RMSE	Root Mean Square Error
RF	Random Forest
UGW	Upstream global wind
UTC	Coordinated Universal Time
XGBoost	eXtreme Gradient Boosting

Nomenclature

Indices

h	Atmospheric layer index
i	Dataset sample index
j	Dataset feature index
k	Layer index in message passing
l	Layer index in MLP
m	Decision tree node index
t	Tree index
ti	Turbine index
(u, v)	Edge from node v to node u in a graph

Sets

$\mathcal{D}$	Dataset used for training machine learning model
$\mathcal{E}$	Set of edges in graph
$\mathcal{N}$	Set of neighbouring nodes
$\mathcal{V}$	Set of nodes in graph

Variables

$\mathbf{A}$	Adjacency matrix
$a_{p,h}$	Hybrid coefficients at atmospheric layer h
A_{rotor}	Rotor swept area (m^2)
a_{vu}	Transformed edge features on edge (v, u)
b	Bias term in neural network

b_h	Hybrid coefficients at atmospheric layer h
c	Constant, with a value of 0.75, used to approximate standard deviation of the wind speed.
C_p	Power coefficient
D	Wake diameter
$d_{xy,vu}$	Euclidean distance between nodes v and u in the horizontal plane.
$d_{z,vu}$	Absolute hub height difference between nodes v and u
$\mathbf{e}_{vu}$	Edge features for edge going from node u to node v
g	gravitational constant ($\mathrm{m\,s^{-2}}$)
G	Global features
g_i	First order gradient statistic for data sample i
H	Number of nodes in hidden layer of MLP
h_i	Second order gradient statistic for data sample i
$\mathbf{h_v}$	Hidden embedding for node v
I_{ref}	Reference turbulence intensity
K	Number of message passing layers
$k_{neighbours}$	Number of neighbours in kNN
M	Number of turbines
m_v	Message aggregated to to node v
N	Number of samples in dataset $\mathcal{D}$
P	Power output (W)
p_h	Pressure at atmospheric layer h
$p_{s,h}$	Surface air pressure at atmospheric layer h
R	Samples in tree parent node
R_{root}	Samples in root node of decision tree
R_L	Samples in left node of decision tree after split
R_R	Samples in right node of decision tree after split
R_{gas}	Specific gas constant ($\mathrm{J\,kg^{-1}\,K^{-1}}$)
R_m	Samples in tree node m
r_{rotor}	Rotor radius (m)
s	Threshold feature
T	Number of decision trees
TI	Wind turbulence intensity
T_{leaf}	Number of leaves in decision tree
T_h	Temperature within atmospheric layer h

t_v	Turbine operational status for node v
$\mathbf{u}_{wind}$	Horizontal wind component towards east
$\mathbf{v}_{wind}$	Horizontal wind component towards north
v_{ws}	Wind speed
w	Weight in neural network
w_{leaf}	Leaf weights in decision tree
wd	Wind direction
$\mathbf{x}$	Input features
$\mathbf{x}_v$	Node features for node v
y	Target
$\hat{y}$	Prediction of target y
α	Wake angle
γ	Yaw angle
γ_{xgb}	Penalty coefficient
Δz_h	Height of atmospheric layer h
ϵ	Constant
η	Learning rate
$\theta_{v,u}$	Relative angle between (v, u) and the wind direction
λ_{xgb}	Penalty coefficient
ρ_{air}	Air density ($\mathrm{kg\,m^{-3}}$)
σ	Standard deviation of wind speed
Ω	Regularization term

Functions

f	Mapping function
$\mathcal{L}$	Loss function
$\mathcal{L}_{RF}$	Random Forest impurity measure
$\mathcal{L}_{split}$	Gain of split for XGBoost
$\mathcal{L}_{XGB}$	Regularized loss function XGBoost
l_{xgb}	Loss function for XGBoost
ϕ	Activation function

Contents

List of Acronyms x

Nomenclature xiii

List of Figures xxi

List of Tables xxv

1 Introduction 1
- 1.1 Background . . . 1
- 1.2 Purpose . . . 4
- 1.3 Scope . . . 4
- 1.4 Limitations . . . 4
- 1.5 Research questions . . . 5

2 Theory 7
- 2.1 Wind Power and Grid Management . . . 7
  - 2.1.1 Wind Power Generation . . . 7
  - 2.1.2 Wake Effects in Wind Farms . . . 8
  - 2.1.3 Grid Management . . . 9
- 2.2 Machine Learning . . . 10
  - 2.2.1 Traditional Machine Learning Methods . . . 10
    - 2.2.1.1 k-Nearest Neighbours Regressor . . . 10
    - 2.2.1.2 Random Forest Regressor . . . 10
    - 2.2.1.3 eXtreme Gradient Boosting Regressor . . . 11
  - 2.2.2 Multi Layer Perceptron . . . 12
- 2.3 Graph Neural Network . . . 13
  - 2.3.1 Graph Representation . . . 13
  - 2.3.2 Message Passing . . . 14

3 Methods 17
- 3.1 Data . . . 17
  - 3.1.1 Supervisory Control and Data Acquisition Data . . . 17
  - 3.1.2 Weather Data, Integrated Forecasting System . . . 18
  - 3.1.3 Forecast Data, Norwegian Meteorological Institute . . . 18
- 3.2 Data Processing . . . 19
  - 3.2.1 Data Cleaning . . . 19

3.2.1.1 Turbine Operation Status . . . 19
3.2.1.2 Removing Curtailed Turbines . . . 20
3.2.1.3 Removing Outliers . . . 20
3.2.1.4 Handling Missing Data . . . 20
3.2.2 Data Splitting and Sampling . . . 20
3.2.3 Normalization . . . 21
3.2.4 Construction of Upstream Global Wind Data . . . 21
3.2.5 Construction of Hourly Data . . . 21
3.3 Graph Representation of the Wind Farm . . . 22
3.3.1 Graph Connections . . . 22
3.3.2 Node Features and Edge Attributes . . . 23
3.4 Baseline Models . . . 23
3.4.1 Power Curve . . . 24
3.4.2 FLORIS . . . 24
3.5 Traditional Machine Learning models . . . 24
3.5.1 Input Feature Representation . . . 25
3.5.2 Training Configuration . . . 25
3.5.2.1 Training Strategy . . . 25
3.5.2.2 Hyperparameter Tuning . . . 25
3.6 Graph Neural Network . . . 26
3.6.1 Architecture . . . 26
3.6.1.1 Encoder . . . 26
3.6.1.2 Processor . . . 26
3.6.1.3 Decoder . . . 26
3.6.2 Training Configuration . . . 27
3.6.2.1 Training Strategy . . . 27
3.6.2.2 Training Settings . . . 28
3.6.2.3 Hyperparameter Tuning . . . 28
3.7 Evaluation Metrics . . . 28
3.8 Testing on Forecast Data . . . 29

4 Results **31**
4.1 Baseline Models . . . 31
4.2 Models Trained on IFS Re-analysis Data . . . 32
4.3 Models Trained on UGW Data . . . 32
4.3.1 Traditional Machine Learning Models . . . 32
4.3.2 Graph Neural Network . . . 34
4.3.2.1 Feature Exploration . . . 34
4.3.2.2 Graph Representation Exploration . . . 34
4.3.3 Multi-Task Learning . . . 36
4.4 Comparative Evaluation of Models . . . 37
4.4.1 Models Trained on 10-Minute Versus 1-Hour Resolution Data 38
4.5 Predicting on Forecast Data . . . 39

5 Discussion **41**
5.1 Models Performance Comparison . . . 41
5.1.1 Baseline Models . . . 41

5.1.2 Traditional Machine Learning Models 42
5.1.3 Graph Neural Network 43
5.2 Effect of Temporal Resolution 44
5.3 Wind Power Forecasting 44
5.4 Global Wind Speed and Wind Direction 45
5.4.1 Sources of Bias in Wind Measurements 45
5.4.2 Global Inflow Conditions 46

6 Conclusion **47**

Bibliography **49**

A Appendix 1 **I**
A.1 Farm Layout and Weather Grid Points I
A.2 Data Cleaning II
A.3 Data Analysis III
A.3.1 North Calibration III
A.3.2 Comparing IFS and UGW wind speed data IV
A.3.3 Wind speed variation IV

List of Figures

1.1 Illustration of different forecasting time frames: very short-term (a few seconds to 30 minutes), short-term (30 minutes to 6 hours), medium-term (6 hours to 1 day), and long-term (more than 1 day). . 2

1.2 Simulated wake losses for a global wind direction of 270 degrees (where 0 degrees corresponds to wind from the north), a global wind speed of 8 m/s, and a turbulence intensity of 10%. This simulation was performed using FLORIS [1] on a randomized layout of a wind farm with 12 turbines. It is assumed that the nacelle of each turbine is oriented perpendicular to the wind direction, and the wake is simulated using the simplified Jensen wake model. 3

2.1 Illustration of a wind turbine showing key features. The left view shows the rotor blades and the rotor diameter as seen from the front. The middle view shows the side profile of the turbine, including the nacelle, hub and hub height. The right view is from above the turbine featuring the yaw angle γ, which is the angle between the rotor axis and the wind direction. 8

2.2 A typical wind turbine power curve illustrating the relationship between wind speed and power output. The curve includes three key thresholds: the cut-in speed where the turbine starts generating power, the rated output speed where the turbine produces maximum power, and the cut-out speed where the turbine shuts down to prevent damage. 9

2.3 Illustration of an MLP neural network. The connections to a single neuron are highlighted in red and illustrating how the output for a neuron is calculated. L denotes the number of hidden layers and H denotes the number of nodes in each layer. 13

2.4 Illustration of a directed graph with node and edge features. Each node v is associated with a feature vector $\mathbf{x}_v$, as indicated for node v_2. Directed edges represent relationships between nodes, with optional edge features such as $\mathbf{e}_{12}$ from v_2 to v_1 shown in red. 14

2.5 Illustration of a message passing function in the first iteration. For node v_5, it receives messages from its neighbours $v \in \mathcal{N}(v_5)$, where each message is computed by applying a ReLU activation to the sum of the neighbouring node's feature $\mathbf{x}_i$ and the edge feature $\mathbf{e}_{5i}$. These messages are aggregated via summation, added to the current node feature $\mathbf{x}_5$, and passed through an MLP to compute the updated node embedding $\mathbf{h}_5^{(1)}$. 15

3.1 Illustration of the wake model for a single turbine. This model was used to determine which turbines are in wake of each other. D is the wake diameter and α is the wake angle. 22

3.2 Steps used to build the graphs. (1) Wake areas were computed for each turbine based on the wind direction (wd). (2) Connections (edges) were added to the graph to represent wake interactions. (3) The clockwise angle $\theta_{v,u}$ between the wind direction and the edge (v, u) was calculated. This angle, $\theta_{v,u}$ was included as an edge attribute, along with other features. Node attributes were also added to the graph. 23

3.3 Overview of the GNN architecture. The wind farm is modelled as a directed graph based on wind direction (wd). Each node (turbine) is encoded using a node encoder MLP that processes local features $\mathbf{x}_v = \{G, t_v\}$, where t_v denotes the turbine's operational status and G the global features. Edge features $\mathbf{e}_{vu}$ include information such as horizontal and vertical distance, and relative angles. These are processed by an edge encoder MLP. The encoded graph is passed through K message-passing layers where node representations $\mathbf{h}_v^{(k)}$ are updated using aggregated messages from neighbouring turbines. Finally, the decoder MLP predicts the output $\hat{y}_v$ for each node v. . . 27

4.1 Models trained using one model per turbine. RMSE in kW for different sets of features. 33

4.2 Models trained using one model for all turbines. RMSE in kW for different sets of features. 33

4.3 Visualization of RMSE per turbine for different graph configurations: fully connected graph, and graphs with wake angles of 3°, 40°, and 85°. 36

4.4 GNN results comparing single-task learning to multi-task learning with additional turbine-level SCADA measurements as output targets, displaying the RMSE. 36

4.5 Absolute error distribution comparison for XGBoost, Best GNN, FLORIS, and Power Curve models for Turbine 3. The average MAE for turbine 3 is shown as a dashed black vertical line. 37

4.6 True vs. Predicted power comparison for XGBoost, Best GNN, FLORIS, and Power Curve models for Turbine 3. 38

4.7 Comparison of models evaluated on forecast data and the re-analysis test data. The x-axis shows the forecast lead times for the forecast data and y-axis the MAE in kW. 39

A.1 Spatial distribution of IFS dataset grid points relative to wind turbine locations. Wind turbine positions are marked in red and grid points from IFS data set in blue [2]. I
A.2 Spatial distribution of MET Norway forecast data grid points used relative to wind turbine locations. Wind turbine positions are marked in red and grid points from MET Norway dataset in burgundy [3]. . . II
A.3 Percentage of time turbine is operational after filtering. III
A.4 Wind direction measurements for all turbines over a randomly selected 150-point time segment, starting at index 800, before and after north calibration. The left plot shows uncalibrated wind direction signals, where significant offsets between turbines are visible. The right plot shows the same measurements after north calibration, with improved alignment between turbines. III
A.5 Comparison of wind speed data from IFS and UGW dataset. The identity line (1:1 line) is shown in black. The windspeed from UGW is generally higher than the wind speed in IFS. IV
A.6 Variability of wind speed from upstream turbines in main wind direction 250°. Main wind direction was given using IFS dataset. V

List of Tables

4.1 Results from the baseline models using IFS input. RMSE and MAE values are given in kW. 31
4.2 Results from the baseline models using UGW input. RMSE and MAE values are given in kW. 32
4.3 Best traditional ML models trained on IFS re-analysis data showing the performance metrics. RMSE and MAE values are given in kW. . 32
4.4 Best models trained on UGW data showing the performance metrics. RMSE and MAE values are given in kW and Farm MAPE in %. . . . 34
4.5 GNN results with different input features, showing the performance metrics. All RMSE and MAE values are given in kW. 34
4.6 Graph connectivity statistics derived from the training dataset for different wake angles. The table presents the average in-degree and percentage of isolated nodes for each turbine (T00-T15). Increasing the wake angle generally enhances connectivity, reducing the number of isolated nodes. 35
4.7 GNN results with different wake angles, showing the performance metrics all in kW except for MAPE. 35
4.8 Comparison of traditional ML models trained on 1-hour and 10-minute resolution data. RMSE values in kW are reported for both resolutions to ensure fair comparison. The best-performing GNN model is included as a reference. 38
4.9 Summary of MAE and RMSE for each model evaluated on both forecast data and re-analysis data from the test set. The metrics are averaged over all forecast lead times. The table includes results for the best-performing ML models (GNN, XGBoost trained on UGW and IFS data), as well as a baseline evaluation using the power curve, which outperforms the other models in terms of both MAE and RMSE when evaluated on the forecast data. 39

A.1 Percentage of data points removed per turbine and cleaning step. Cleaning steps are applied sequentially, meaning each step operates on the remaining data from the previous step. The total row represents the cumulative percentage of invalid data points per turbine. II

1
Introduction

The increasing global demand for energy, coupled with the urgent need to address climate change, has accelerated the adoption of renewable energy sources. Among these, wind power has emerged as one of the most promising solutions due to its sustainability, widespread availability, and zero carbon emissions [4]. However, the inherent intermittency and variability of wind make accurate wind power forecasting a critical challenge for grid integration, scheduling, and energy market participation [5]. Machine learning (ML) is advancing and could be used to address the complexities of wind power forecasting [6]. By leveraging extensive datasets, ML models can find non-linear patterns between meteorological variables and wind energy output, leading to more accurate and efficient predictions compared to traditional methods.

In this thesis, different ML methods for wind power prediction are compared and evaluated. Wind power prediction refers to the real-time estimation of power output based on current conditions, while wind power forecasting refers to estimating future power generation over short to long time horizons. The study examines the impact of different input features and compares models of varying complexity, ranging from simple algorithms like k-Nearest Neighbours (kNN) regression to more advanced architectures such as a graph neural network (GNN). Finally, the ability of these models to generalize from prediction to wind power forecasting is assessed.

This chapter introduces the background and motivation for the study, defines key concepts related to wind power prediction and forecasting, and outlines commonly used methods in the field. It concludes with the purpose of the thesis, its scope, its limitations, and presents the research questions.

1.1 Background

Sweden aims to achieve 100% renewable energy production by 2040, as a part of its commitment to a sustainable and clean energy future [7]. As of 2024, wind power accounts for approximately 25% of Sweden's total electricity production. The goal is to increase this share to 34% by 2030 and to 50% by 2050 according to the Swedish Wind Energy Association [8]. At the same time, The Swedish Energy Agency projects Sweden's electricity demand to grow from 134 TWh in 2020 to between 200 and 270 TWh in 2040, depending on future developments [9]. This rapid expansion makes accurate wind power forecasting crucial, as it plays a vital role in mitigating challenges, such as grid instability, associated with the stochastic nature of wind energy [4].

Furthermore, accurate forecasts are also important to maximize profitability for

wind farm owners. They often enter long-term contracts committing to deliver a specific amount of energy at a fixed price [10]. If actual production falls short, wind farm owners must purchase energy from the market to cover the shortfall. This can also lead to additional costs through balance settlement, where discrepancies between forecasted and actual production result in financial penalties [11].

The power generated by wind turbines is influenced by several factors, including wind speed and direction, topology and seasonal variations [4, 12]. Additionally, wake effects impact the power production and occurs because downstream turbines experience a reduction in available energy due to the velocity deficit created by upstream turbines' wakes, see Figure 1.2. Previous research has shown that to accurately predict power generation of wind farms the temporal and spatial dynamics need to be captured in the model [13].

Traditional wind power forecasting methods include both physical and statistical models, each with its own advantages depending on the forecasting horizon: very short-term (up to 30 minutes), short-term (up to 6 hours), medium-term (up to 24 hours), and long-term (beyond 24 hours), as shown in Figure 1.1. One of the most widely used physical methods is Numerical Weather Prediction (NWP). It relies on integrated hydro- and thermodynamic models to simulate weather conditions for accurately forecasting wind speed [13]. The predicted wind speed is then mapped to the wind farms power curve to estimate the power output. While NWP models perform well for long-term forecasting, their reliance on complex physical data can lead to high computational costs, making them less suitable for short-term forecasting applications. Furthermore, the estimation accuracy is sufficient for larger regional areas, where local variations such as topology tend to average out, but inadequate on a wind farm level due to the influence of site-specific factors [14]. Moreover, since the wind farm power curve does not account for wake effects, using NWP models for forecasting can lead to inaccuracies in power predictions.

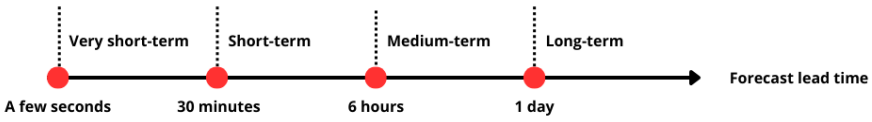

Figure 1.1: Illustration of different forecasting time frames: very short-term (a few seconds to 30 minutes), short-term (30 minutes to 6 hours), medium-term (6 hours to 1 day), and long-term (more than 1 day).

Simplified physical models of wake effects on a wind farm level have been developed such as Flow Redirection and Induction in steady-state (FLORIS) by National Renewable Energy Laboratory (NREL) [1]. The FLORIS model can be used to estimate long-term power production, such as annual energy production (AEP). It can also help identify optimal control settings and analyse performance across different wind directions. Figure 1.2 shows an example of wakes created from wind turbines simulated using FLORIS. However, the FLORIS model is a simplified engineering wake model that assumes steady-state dynamics for the wake. Additionally, it provides time-averaged results, which makes it unsuitable for both medium-term and

shorter-term forecasting.

Statistical methods have also been developed for wind power prediction. They offer a lower computational cost than physical methods, but they struggle to capture the complex, non-linear relationships between environmental factors (like wind speed, temperature, and turbulence) and power output [4]. One of the most used statistical models for wind power forecasting is the Autoregressive (AR) model. The AR model is a time series method where the future value of a series depends linearly on its historical values [15]. To improve on this, the Autoregressive Moving Average (ARMA) model extends the AR model by incorporating past forecast errors into the prediction. This helps to adjust for random fluctuations in the data and can improve forecast accuracy. While both AR and ARMA models perform well for very short-term forecasting, they face challenges when it comes to medium- and long-term predictions [4].

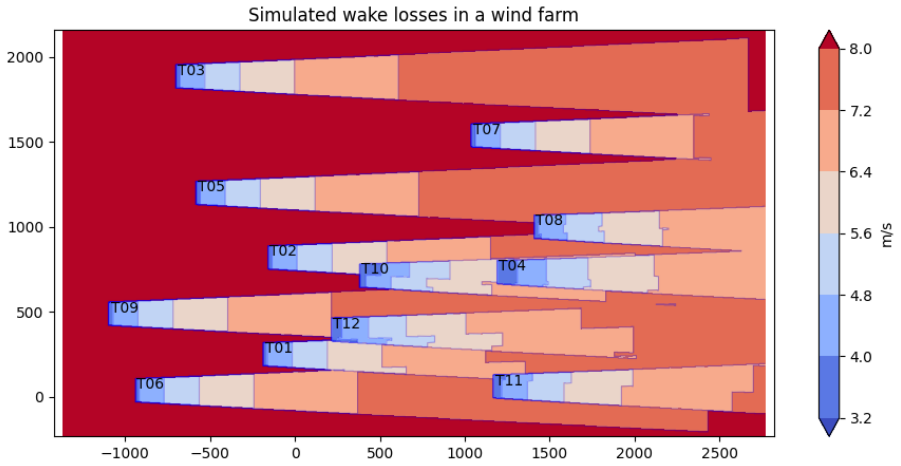

Figure 1.2: Simulated wake losses for a global wind direction of 270 degrees (where 0 degrees corresponds to wind from the north), a global wind speed of 8 m/s, and a turbulence intensity of 10%. This simulation was performed using FLORIS [1] on a randomized layout of a wind farm with 12 turbines. It is assumed that the nacelle of each turbine is oriented perpendicular to the wind direction, and the wake is simulated using the simplified Jensen wake model.

With advancements in artificial intelligence (AI) and access to more data, AI based methods have emerged as a powerful data driven tool for wind power prediction [14, 4]. Supervisory Control and Data Acquisition (SCADA) data, collected from wind turbines, plays a crucial role in these advancements by providing high-resolution historical data on wind speeds, power output, turbine performance, and environmental conditions at the turbine level [14].

Traditional ML models, such as eXtreme Gradient Boosting (XGBoost) regression and Random Forest (RF), have demonstrated high performance and significant promise in prediction and forecasting accuracy of wind power [5]. Unlike simpler statistical models used for wind power prediction, ML models can capture non-linear relationships in historical data. GNNs have also been shown to capture wake effects in simulated wind farm data [12]. One key advantage of GNNs over traditional ML

methods is their ability to model the relationships between neighbouring turbines more effectively, capturing both spatial and temporal dependencies [5].

Furthermore, GNNs combined with Long Short-Term Memory (LSTM) networks have been found to outperform traditional statistical methods in very short-term to short-term wind power forecasting using time-series SCADA data [14]. Other deep learning (DL) architectures, such as transformers, have also shown effective in handling time-series data for forecasting short- to medium-term wind farm power output [16].

1.2 Purpose

This thesis evaluates various ML models for wind power prediction, focusing on the impact of different input features, graph representations and model accuracy. Additionally, it assesses the models' potential to extend from power prediction to wind power forecasting using forecasted input data.

1.3 Scope

This thesis presents a comparative analysis of a GNN and traditional ML models for short- to medium-term forecasting. To develop these models, historical SCADA data from a single onshore wind farm in Sweden is utilized, in combination with historical re-analysis weather data corresponding to the wind farm's location. The models' performance is evaluated against two baseline approaches: (i) traditional power curve-based prediction methods and (ii) simulations conducted using the FLORIS wake model.

1.4 Limitations

This thesis and the developed models have certain limitations that should be acknowledged. A significant limitation is the availability of SCADA data for the selected wind farm with 16 wind turbines, which spans only a five-year period and contains noise, potentially affecting the generalizability of the results. Additionally, the sensors collecting the data are located behind the turbines and measure only a single point, which does not accurately represent the flow field passing through the rotors. Furthermore, the data cleaning process may introduce biases or exclude potentially valuable information, which could influence the model's predictions.

Another constraint concerns the availability and quality of global weather data used in the model. Since no measured historical weather data at hub height is available for the wind farm, and historical re-analysis weather datasets suffer from low spatial and temporal resolution. SCADA data was used to approximate global wind conditions. However, the measurements are taken behind the turbines, where flow is disturbed. The sensor data therefore requires processing, and the wind direction measurements are not north calibrated, which can introduce further inaccuracies in the wind conditions data. Furthermore, the model does not incorporate the uncer-

tainty of weather forecasts, which are often unreliable due to the complexity of wind dynamics, or the potential discrepancies between forecast and training data.

Due to time constraints, some hyperparameters within the models were fixed rather than thoroughly optimized. This limitation may affect the models' overall performance. Similarly, the number of ML models tested was constrained by time, which restrict the exploration of potentially more effective model architectures or configurations.

1.5 Research questions

The following research questions are answered in this thesis.

1. **Model Performance and Comparison**
   - How do traditional ML models (XGBoost, RF, kNN, MLP) compare to GNN in terms of accuracy for wind power prediction and how do they generalize to wind power forecasting?
   - What are the advantages and limitations of using a graph-based approach to model wind farm interactions?
2. **Feature Importance**
   - What input features are most influential in achieving accurate wind power predictions across different models?
   - How does feature importance differ between traditional ML models and GNN?
3. **Multi-Task Learning for Enhanced Predictions**
   - Can incorporating additional outputs, such as local wind speed and turbulence intensity at each turbine, improve power prediction performance in a multi-task learning setting?

2
Theory

This chapter covers the theory behind wind power generation. It explains the factors influencing turbine performance, such as wind speed and turbulence, as well as the concept of wake effects in wind farms. Additionally, the chapter introduces ML models, describing the traditional ML models used as well as GNNs.

2.1 Wind Power and Grid Management

Wind power plays a vital role in renewable energy systems, but its integration into the grid requires careful management to ensure stability and reliability [5]. This section covers wind power generation, wake effects in wind farms, and the importance of grid management in optimizing renewable energy resources.

2.1.1 Wind Power Generation

Wind turbines convert kinetic energy from wind into electrical energy, where inflow conditions, including wind speed and direction, are the primary factors influencing the amount of power generated [17]. The power output of a wind turbine is influenced by several other factors, including air temperature and air density. In addition, variability in wind power generation is affected by dynamic atmospheric processes occurring at different spatial and temporal scales, such as surface-layer and boundary-layer turbulence [18]. These variables affect wind flow characteristics and consequently influence the turbine's operational efficiency.

Turbulence intensity (TI) refers to the degree of fluctuation in wind speed relative to its mean value. It is mathematically expressed as:

$$\mathrm{TI} = \frac{\sigma}{\bar{v}_{\mathrm{ws}}} \tag{2.1}$$

where σ represents the standard deviation of wind speed, and $\bar{v}_{\mathrm{ws}}$ denotes the mean wind speed. Turbulence intensity is a critical parameter affecting both turbine performance and structural loading. For instance, a higher turbulence intensity can induce additional mechanical fatigue on turbine components.

The nacelle, which houses the generator and other essential components, is mounted on a yaw system that enables it to rotate and align with the prevailing wind direction [19]. Proper alignment maximizes energy capture by ensuring that wind energy is effectively converted into electrical power. The nacelle is positioned at hub height, defined as the vertical distance from the ground to the centre of the rotor to which the blades are attached, see Figure 2.1. Power generation may also be regulated by

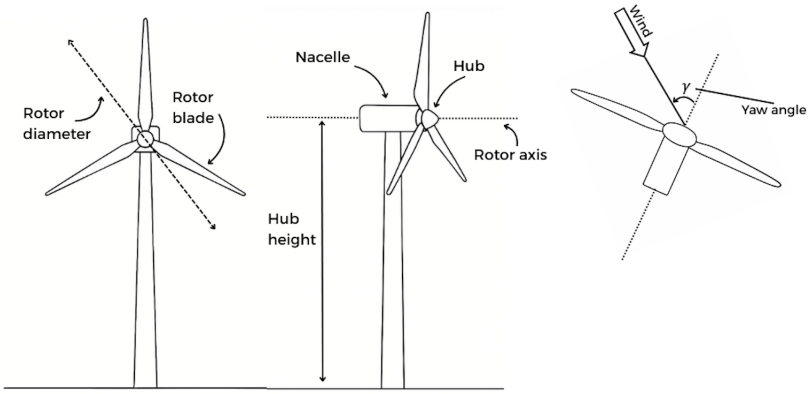

Figure 2.1: Illustration of a wind turbine showing key features. The left view shows the rotor blades and the rotor diameter as seen from the front. The middle view shows the side profile of the turbine, including the nacelle, hub and hub height. The right view is from above the turbine featuring the yaw angle γ, which is the angle between the rotor axis and the wind direction.

the blade pitch, which refers to the angle between each blade and the rotor's plane of rotation. Adjusting the blade pitch controls the amount of aerodynamic force converted into rotational energy, directly influencing the amount of power produced.

The power output of a wind turbine is typically represented by a power curve, which illustrates a non-linear relationship between wind speed and electrical power output, as shown in Figure 2.2. This relationship is described by the aerodynamic power equation:

$$P = \frac{1}{2}\rho_{air} A_{rotor} v_{\text{ws}}^3 C_p \tag{2.2}$$

where P is the power output (Watts), ρ_{air} is the air density (kg/m^3) and $A_{rotor} = \pi r_{rotor}^2$ is the swept area of the rotor (m^2), where r_{rotor} is the rotor radius (m), v_{ws} is the wind speed (m/s), C_p is the power coefficient, representing the turbine's efficiency [20]. As indicated by this equation, the power output is directly proportional to the square of the rotor radius (r_{rotor}^2) and the cube of the wind speed (v_{ws}^3) [21]. The cut-in wind speed is the minimum wind speed at which the turbine starts to generate power, while the rated wind speed is the wind speed at which the turbine produces its maximum rated power [17]. The cut-out wind speed refers to the maximum wind speed beyond which the wind turbine ceases operation to prevent mechanical damage. It is important to note that the power curve represents the power output for a single turbine and does not account for wake effects or interactions between turbines at the farm level.

2.1.2 Wake Effects in Wind Farms

In wind farm operations, upstream turbines are exposed to the incoming wind and generally receive undisturbed airflow. In contrast, downstream turbines operate within the wake generated by upstream turbines. A wake refers to a flow region characterized by reduced wind speed and increased turbulence intensity, resulting

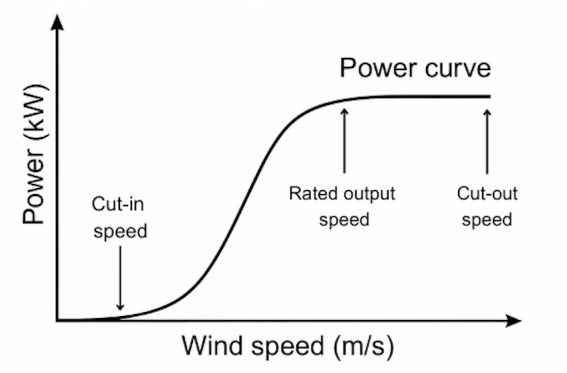

Figure 2.2: A typical wind turbine power curve illustrating the relationship between wind speed and power output. The curve includes three key thresholds: the cut-in speed where the turbine starts generating power, the rated output speed where the turbine produces maximum power, and the cut-out speed where the turbine shuts down to prevent damage.

from the extraction of energy by the upstream turbine. This disturbed flow can negatively affect both the power output and structural loading of downstream turbines [22]. As shown in Figure 1.2, turbine T06 is located upstream relative to the wind direction, while turbine T11 lies downstream, within the combined wake influence of both turbines T06 and T01. Wind farms typically have a dominant wind direction, meaning some turbines consistently operate upstream with undisturbed airflow, while others are frequently exposed to wake effects.

The wake typically extends approximately 8 to 10 rotor diameters downstream of a turbine, but it may extend even further, influencing the wind conditions experienced by downstream turbines [23]. A higher turbulence intensity leads to a faster wake recovery than a lower turbulence intensity. Adjusting the yaw angle can also shift the wake's position, altering its effect on other turbines [19].

Various engineering models have been developed to represent wake behaviour in wind farms. The Jensen wake model is a widely adopted analytical approach assuming the wake expands linearly with a constant expansion rate [22]. Figure 1.2 illustrates a wind farm layout in which the Jensen model is applied to simulate wake effects.

2.1.3 Grid Management

Since the inflow conditions over a wind farm often fluctuate, understanding the temporal variability of wind speed at the farm level is crucial for having a high and stable wind power generation [24]. Wind tends to become more predictable with increasing height above ground, while complex terrain introduces greater uncertainty and reduces the predictability of inflow conditions.

As a result, grid management becomes a crucial component of modern electricity systems. The stability and reliability of the power grid rely on the continuous balance between electricity supply and demand [25]. Imbalances, whether caused by oversupply, shortages, or fluctuations from renewable sources such as wind, may result in grid instability, congestion, or even blackouts.

One strategy employed to maintain grid stability is the Frequency Containment Reserve for Disturbances (FCR-D), regulating the grid frequency within acceptable limits [26]. Specifically, FCR-D down is used to respond to disturbances or imbalance in the grid by reducing the power output from certain power plants, including wind farms. This reduction causes the generated power to differ from the expected power production of the wind turbines, which makes short- to medium-term power forecasting more difficult.

2.2 Machine Learning

This section provides an overview of ML and the models utilized in this study, categorized into traditional ML methods and GNNs.

Certain ML methods, such as kNN regressor, rely more heavily on manually crafted input features, often selected based on domain knowledge and the objective in mind. In contrast, other models like MLP or GNNs can automatically learn and extract useful feature representations directly from the data [27].

In supervised learning a model is trained using a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ of size N, where $\mathbf{x}_i$ is the feature vector and y_i the target variable for the i-th sample. The objective is to minimize the loss function ($\mathcal{L}$) by finding a mapping function, $f(\mathbf{x}_i) \approx y_i$ that maps the input to the output [28]. The exact form of the loss function varies depending on the model or the specific optimization goal.

2.2.1 Traditional Machine Learning Methods

In this subsection the theory behind the traditional ML models is presented. Except for MLP, these models are all non-parametric, meaning they do not assume a fixed number of parameters or a specific data distribution [29]. This allows them to be flexible and adapt to a wide range of tasks.

2.2.1.1 k-Nearest Neighbours Regressor

kNN Regressor predicts the output for a given unseen data point by averaging the target values for the k-nearest neighbours [30]. The neighbours are determined using a distance metric such as Euclidean distance. The nearest neighbours used are the data points which are part of the training data of the model. Hence, this method requires no optimization only determination of distance metric and the number of neighbours ($k_{neighbours}$).

2.2.1.2 Random Forest Regressor

RF regressor is an ensemble learning method that constructs multiple decision trees and aggregates their outputs [31]. The objective is to create diverse trees while reducing the variance in the model's predictions, thereby improving the model's generalization capability.

Each tree in the ensemble is trained on a randomly sampled subset of the training data. At each node of a decision tree, a split is created by selecting the optimal

feature and a threshold for that feature that best separates the data into left and right child nodes [31]. Rather than evaluating all features, only a randomly chosen subset is considered at each split to introduce additional randomness. The best split is chosen by minimizing the weighted sum of variances in the child nodes. That is, the variance of the target values within each child node, weighted by the number of samples they contain. Given a split feature j at threshold s, the impurity measure $\mathcal{L}_{RF}(j, s)$ is defined as:

$$\mathcal{L}_{RF}(j, s) = \sum_{m \in \{L,R\}} \frac{|R_m|}{|R|} \cdot \mathrm{Var}(y \mid R_m) \tag{2.3}$$

where $|R_m|$ is the number of samples in tree node m, $|R|$ is the total number of samples in the tree parent node.

Every decision tree recursively applies such splits to partition the input space, meaning the space of all possible feature combinations, into distinct regions R_m. The prediction for a new input $\mathbf{x}$ is then computed by averaging the target values y_i of all training samples $\mathbf{x}_i$ that fall into the same region R_m as $\mathbf{x}$:

$$\hat{y}(\mathbf{x}) = \frac{1}{|R_m|} \sum_{\mathbf{x}_i \in R_m} y_i. \tag{2.4}$$

For regression, the final prediction $\hat{y}$ for an input $\mathbf{x}$ is computed by averaging the predictions from all T decision trees:

$$\hat{y}(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^{T} \hat{y}_t(\mathbf{x}) \tag{2.5}$$

where $\hat{y}_t(\mathbf{x})$ is the predicted value for $\mathbf{x}$ from tree t.

2.2.1.3 eXtreme Gradient Boosting Regressor

XGBoost regressor is a tree-based ensemble method which uses T additive functions or trees to make a prediction of the output [32]. The objective is to additively minimize the loss $\mathcal{L}_{XGB}$:

$$\mathcal{L}_{XGB}^{(t)} = \sum_{i=1}^{N} l_{xgb}(y_i, \hat{y}_i^{(t-1)} + f_{tree,t}(\mathbf{x}_i)) + \Omega(f_{tree,t}) \tag{2.6}$$

where l_{xgb} is a convex differentiable loss function that measures the difference between the prediction $\hat{y}_i$ and the target y_i. For each iteration t, a tree $f_{tree,t}$ that minimizes the loss is constructed. $\Omega(f_{tree,t})$ is a regularization term penalizing complexity in the trees:

$$\Omega(f_{tree,t}) = \gamma_{xgb} T_{leaf} + \frac{1}{2} \lambda_{xgb} \sum_{m=1}^{T_{leaf}} ||w_{leaf,m}||^2 \tag{2.7}$$

where T_{leaf} is the number of leaves in the tree. γ_{xgb} ($\gamma_{xgb} > 0$) and λ_{xgb} ($\lambda_{xgb} \geq 0$) are penalty terms, while w_{leaf} are the leaf weights of the tree.

An initial guess ($\hat{y}^0$), such as the average value of all targets, is used [32]. From this guess the residuals are calculated and a tree $f_{tree,t}$ is constructed to minimize

the residuals of that initial guess by iteratively and greedily adding branches to the tree. To minimize the loss a second order Taylor approximation can be used. Let R_{root} denote the root tree and R_L and R_R denote the left and right node after a split and $g_i = \partial_{\hat{y}^{t-1}} l_{xgb}(y_i, \hat{y}^{(t-1)})$ is the first order gradient statistic of the loss function and $h_i = \partial^2_{\hat{y}^{t-1}} l_{xgb}(y_i, \hat{y}^{(t-1)})$ the second. Then the gain ($\mathcal{L}_{split}$) or reduction in loss is given by:

$$\mathcal{L}_{split} = \frac{1}{2}\Big[\frac{(\sum_{i\in R_L} g_i)^2}{\sum_{i\in R_L} h_i + \lambda_{xgb}} + \frac{(\sum_{i\in R_R} g_i)^2}{\sum_{i\in R_R} h_i + \lambda_{xgb}} - \frac{(\sum_{i\in R_{root}} g_i)^2}{\sum_{i\in R_{root}} h_i + \lambda_{xgb}}\Big] - \gamma_{xgb} \tag{2.8}$$

The best split is the one maximizing the gain ($\mathcal{L}_{split}$) [32]. Negative values in gain mean no gain and the construction of the tree would stop. Otherwise, the tree is constructed by repeatedly creating and evaluating splits greedily until stopping criterion, max depth or all leafs have one residual.

Once the tree $f_{tree,t}$ has been constructed the predicted output value for each region m (i.e., each leaf), referred to as the leaf weight ($w_{m,leaf}$) is calculated as:

$$w_{m,leaf} = -\frac{\sum_{i\in R_m} g_i}{\sum_{i\in R_m} h_i + \lambda_{xgb}}. \tag{2.9}$$

The new prediction is then calculated according to:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta \cdot f_{tree,t}(\mathbf{x}_i) \tag{2.10}$$

where η is the learning rate and $f_{tree,t}(\mathbf{x}_i)$ returns the leaf weight corresponding to the region that $\mathbf{x}_i$ belongs to. The residuals are then recalculated, and a new tree is fitted to minimize the updated residuals [32]. The construction of trees $f_{tree,t}$ is then repeated until the specified number of trees has been constructed. The constructed trees then constitute the model together with the initial guess. The final prediction, $\hat{y}$, is the sum of the initial prediction, $\hat{y}^0$, and the weighted sum of the output values from the constructed trees, where each tree's output is scaled by the learning rate η:

$$\hat{y}(\mathbf{x}) = \hat{y}^0 + \eta \sum_{t=1}^{T} f_{tree,t}(\mathbf{x}) \tag{2.11}$$

2.2.2 Multi Layer Perceptron

A perceptron is the fundamental building block of a neural network [33]. It computes a weighted sum of its inputs, followed by an activation function ϕ to introduce non-linearity:

$$y = \phi(\sum_i w_i x_i + b) \tag{2.12}$$

where w_i are the weights applied to each input feature x_i, and b is a bias term. There are several different activation functions and a commonly used one is the rectified linear unit (ReLU) function:

$$\text{ReLU}(\cdot) = \max(0, \cdot) \tag{2.13}$$

The ReLU function takes an input and outputs the maximum between zero and the input, setting all negative values to zero.

A MLP consists of several perceptrons stacked together in different layers, see Figure 2.3. L denotes the number of layers and H the number of perceptrons in each layer. Both weights ($\mathbf{w}$) and biases ($\mathbf{b}$) are learnable parameters optimized through the loss function using back propagation during training. The final layer produces the network's prediction ($\hat{y}$).

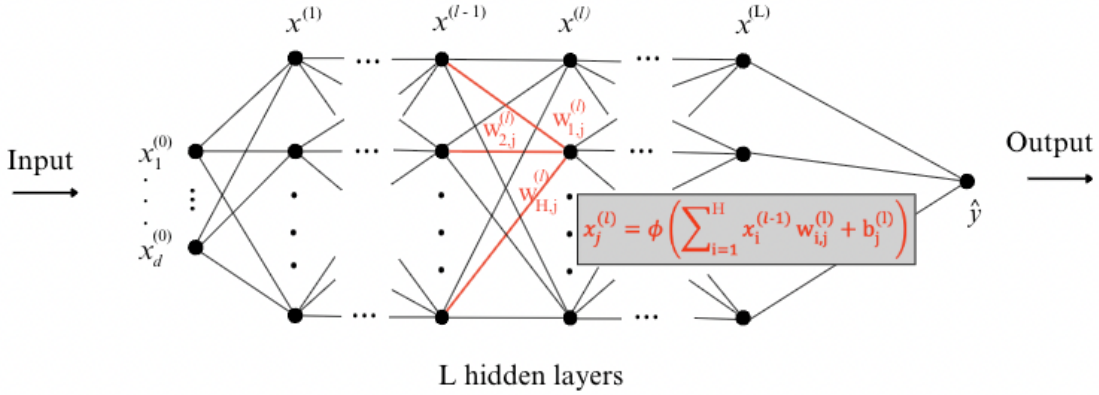

Figure 2.3: Illustration of an MLP neural network. The connections to a single neuron are highlighted in red and illustrating how the output for a neuron is calculated. L denotes the number of hidden layers and H denotes the number of nodes in each layer.

2.3 Graph Neural Network

GNNs are DL models specifically designed to process graph-structured data. The following section introduces the fundamental concepts of graphs and the message passing mechanism, which enables GNNs to capture relationships between nodes.

2.3.1 Graph Representation

A graph is a data structure consisting of a set nodes ($\mathcal{V}$) and a set edges ($\mathcal{E}$) [34]. Each node represents an entity, while the edges represent the relationships or connections between those entities. An edge from node $u \in \mathcal{V}$ to node $v \in \mathcal{V}$ is denoted as $(v, u) \in \mathcal{E}$. In a directed graph the direction of the edge from u to v carries meaning as the edges can have different attributes depending on the direction. An illustration of such a graph is shown in Figure 2.4. Edges can have several attributes, often referred to as edge features, denoted as $\mathbf{e}_{vu}$, which represents the feature associated with the directed edge from node u to node v. These features encode various relationship properties between the nodes such as connection strength, distance, or other contextual information.

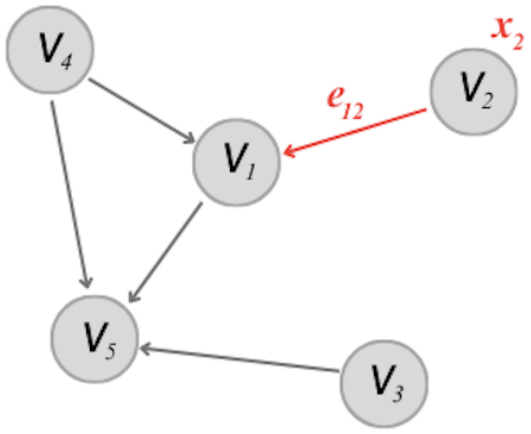

Figure 2.4: Illustration of a directed graph with node and edge features. Each node v is associated with a feature vector $\mathbf{x}_v$, as indicated for node v_2. Directed edges represent relationships between nodes, with optional edge features such as $\mathbf{e}_{12}$ from v_2 to v_1 shown in red.

Nodes can have features as well. These are specific to the node and denoted as $\mathbf{x}_v$. Node features capture characteristics specific to each node [34].

A graph can be represented by an adjacency matrix $\mathbf{A} = \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$, where one node represents each index in the matrix. A graph is undirected if $\mathbf{A}$ is a symmetric matrix, i.e $\mathbf{A} = \mathbf{A}^T$.

$$\mathbf{A}[u,v] = \begin{cases} 1 & \text{if } (u,v) \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.14}$$

For sparse graphs a more compact representation of a graph is an edge list, which consists of the set of edges in the graph [34]. Several useful statistical properties can be calculated from a graph such as in-degree, which is the number of edges connected to a node, and out-degree, which is the number of edges originating from a node. A node is isolated if both its in-degree and out-degree is 0. Notably, nodes in a graph are not independent and identically distributed, as their properties and connections are influenced by the structure of the graph.

2.3.2 Message Passing

One fundamental concept in GNN architecture is message passing, which enables information exchange between connected nodes in a graph [34]. Each node $v \in \mathcal{V}$ has a hidden embedding $\mathbf{h}_v^{(k)}$ that is updated at each iteration k. This update is based on information aggregated from the node's neighbourhood $\mathcal{N}(v)$ using a differentiable aggregation function:

$$m_v^{(k)} = \text{AGGREGATE}\left(\{\mathbf{h}_u^{(k)}, \forall u \in \mathcal{N}(v)\}\right) \tag{2.15}$$

where $m_v^{(k)}$ is the aggregated message to node v at iteration k and AGGREGATE$(\cdot)$ defines how the information from neighbouring nodes is combined. The initial node

embeddings at layer $k = 0$ are set to the input features of each node, i.e., $\mathbf{h}_v^{(0)} = \mathbf{x}_v$, for all $v \in \mathcal{V}$.

The node's hidden embedding is updated by applying another differentiable update function that integrates both the aggregated neighbourhood information and the node's previous state:

$$\mathbf{h}_v^{(k+1)} = \text{UPDATE}\left(\mathbf{h}_v^{(k)}, m_v^{(k)}\right) \tag{2.16}$$

where UPDATE$(\cdot)$ determines how the node's own representation is refined based on the aggregated information.

After running K layers (i.e. iterations) of message passing, the final node embeddings are obtained from the last layer. $\mathbf{h}_v^{(K)}$ represents the final embedding for node v, capturing both its own features and the information propagated from its neighbours through K layers of message passing.

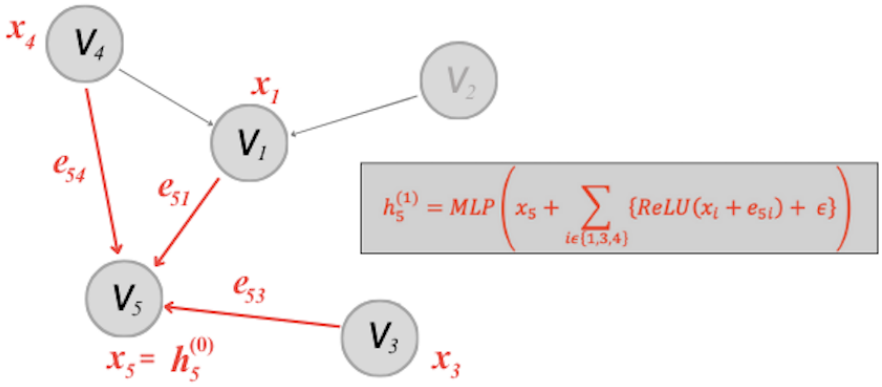

Figure 2.5: Illustration of a message passing function in the first iteration. For node v_5, it receives messages from its neighbours $v \in \mathcal{N}(v_5)$, where each message is computed by applying a ReLU activation to the sum of the neighbouring node's feature $\mathbf{x}_i$ and the edge feature $\mathbf{e}_{5i}$. These messages are aggregated via summation, added to the current node feature $\mathbf{x}_5$, and passed through an MLP to compute the updated node embedding $\mathbf{h}_5^{(1)}$.

In the case of GENeralized Graph Convolution (GENConv), message passing follows a generalized aggregation framework where both node and edge features can be incorporated [35]. The aggregated message is computed as a learned transformation of neighbouring node embeddings and their connecting edge features. Specifically, messages to node v are formed by applying the non-linear activation function ReLU to the sum of the neighbours hidden embedding $\mathbf{h}_u^{(k)}$ and the edge features $\mathbf{e}_{vu}$ for all neighbours u in $\mathcal{N}(v)$. These transformed messages are then aggregated using a aggregation function, such as softmax, power mean, summation, depending on the chosen configuration. An example of a GENConv message passing at the first iteration, with summation aggregation is shown in Figure 2.5. The message passing function is given as:

$$\mathbf{h}_v^{(k+1)} = \mathrm{MLP}\left(\mathbf{h}_v^{(k)} + \mathrm{AGGREGATE}\left(\{\mathrm{ReLU}(\mathbf{h}_u^{(k)} + \mathbf{e}_{vu}) \mid u \in \mathcal{N}(v)\} + \epsilon\right)\right). \tag{2.17}$$

where MLP is the update function composed of linear transformations, non-linear activations (such as ReLU), and optionally layer normalization, and ϵ is a constant. The MLP refines the combined information from the node's previous state and its aggregated neighbourhood messages.

3
Methods

This chapter describes the methodological approach employed in developing and evaluating the wind power prediction models. It outlines the data sources and preprocessing steps used to construct the input datasets. The development of baseline models is presented, followed by a detailed explanation of the design and configuration of both the traditional ML models and the GNN, including the training strategies and hyperparameter tuning processes. Finally, the evaluation metrics used to assess model performance are introduced.

3.1 Data

The following section outlines the datasets used for training and for evaluating the ML models. The data includes sensor measurements, historical re-analysis weather data and forecast data, all in UTC time.

3.1.1 Supervisory Control and Data Acquisition Data

The SCADA system continuously monitors and records operational and environmental parameters from each wind turbine within the wind farm. The wind farm consists of 16 turbines, each with a rated capacity of 3800 kW. The sensors are positioned behind the rotor, measuring wind conditions downstream of the blades. However, the wind speed is corrected for being located downstream of the rotor, after blade interaction, by using a transfer function. This function is determined by the original equipment manufacturer and is kept secret. The manufacturer can change the transfer function at any time without notifying the farm owner or operators.

The dataset was provided by a wind farm owner and consists of 10-minute averaged sensor readings over a 5-year time frame from 2020 to the end of 2024. The SCADA dataset includes the following parameters:

- Wind speed (m/s): Providing localized wind speed corrected for being behind the turbine using transfer function.
- Wind speed standard deviation: (m/s): Represents the wind speed variability, corrected for being behind the turbine using transfer function. Serves as an indicator of turbulence intensity.
- Wind direction ($degrees$): Indicating the direction of the wind relative to true north, defined as 0°.
- Rotor speed (RPM): Representing the rotational speed of the rotor.
- Power output (kW): Refers to the electrical power generated by each turbine.

- Nacelle position (*degrees*): Indicates the yaw angle, representing the orientation of the nacelle relative to true north. This angle is used to optimize turbine alignment with the dominant wind direction.
- Blade pitch angle (*degrees*): The angle of the blades relative to the rotor plane, controlling how much wind energy is captured.
- Curtailment mode: An operational parameter that classifies turbine derating into 13 distinct curtailment categories, each corresponding to specific limitations in power output due to regulatory, mechanical, or operational constraints. Mode 0 represents normal operation, and the other modes are different reasons for derated operation. Mode 18 is different than the others and is used when turbines operate under an external setpoint, such as during power plant control events like FCR-D. The turbine control software frequently sets mode 18 even during normal operation.

3.1.2 Weather Data, Integrated Forecasting System

The wind farm data provides local measurements from individual turbines. To incorporate additional weather variables, historical weather data from Open-Meteo's API [2] was used. This API sources its data from the European Centre for Medium-Range Weather Forecasts (ECMWF). Open-Meteo provides historical re-analysis datasets from ECMWF's Integrated Forecasting System (IFS), offering a spatial resolution of approximately 9 km, see Figure A.1 in section A.1 in Appendix A. The following meteorological variables were extracted from the IFS dataset:

- Wind speed at 100 m above ground (m/s)
- Wind direction at 100 m above ground (*degrees*)
- Temperature at 2 m above ground ($°C$)
- Relative humidity at 2 m (hPa)

Since the IFS provides data at an hourly temporal resolution, interpolation was necessary to align it with the 10-minute resolution of the SCADA data and turbine measurements. Wind direction was first decomposed into its sine and cosine components to preserve angular continuity, followed by linear interpolation. Wind speed, temperature, and relative humidity were also linearly interpolated.

3.1.3 Forecast Data, Norwegian Meteorological Institute

Historical forecast data was extracted from the Norwegian Meteorological Institute [3]. The data has a spatial resolution of 2.5 km and a temporal resolution of 1 hour, with new forecasts generated every 6 hours [36], see Figure A.2 in section A.1 in Appendix A.

To obtain the wind speed and wind direction at the hub height of the turbines firstly the pressure (p_h) at each model level was calculated from the hybrid coordinate system:

$$p_h = a_{p,h} + b_h \cdot p_{s,h} \tag{3.1}$$

where $p_{s,h}$ is the surface air pressure at atmospheric layer h and $a_{p,h}$ and b_h are

hybrid coefficients at atmospheric layer h. The height (Δz_h) of each layer was then calculated using the hydrostatic equation and the ideal gas law:

$$\Delta z_h = \frac{R_{gas} \cdot T_h}{g} \cdot ln(\frac{p_{h+1}}{p_h}) \tag{3.2}$$

where $R_{gas} = 287\ \mathrm{J\,kg^{-1}\,K^{-1}}$ denotes the specific gas constant for dry air, and $g = 9.81\ \mathrm{m\,s^{-2}}$ the gravitational constant. T_h denotes the temperature within the atmospheric layer h. p_h is the pressure at the bottom of the layer and p_{h+1} at the top of the layer. The height above ground for each model level was then calculated by cumulatively summing the layer heights.

Following the estimation of height levels, the horizontal wind components $\mathbf{u}_{wind}$ towards east and $\mathbf{v}_{wind}$ towards north were linearly interpolated between the two closest heights to the hub height of the turbines. These interpolated values were then spatially interpolated to the geographic location of the turbines. Lastly, the wind speed (v_{ws}) was calculated from the interpolated $\mathbf{u}_{wind}$ and $\mathbf{v}_{wind}$ wind components using the following equation:

$$\mathrm{wd} = \left(180^\circ + \frac{180^\circ}{\pi} \cdot \tan^{-1}\left(\frac{\mathbf{u}_{wind}}{\mathbf{v}_{wind}}\right)\right) \bmod 360^\circ. \tag{3.4}$$

Additionally, the air temperature was extracted at the vertical level closest to 2 meters above ground and converted to degrees Celsius. This was done to ensure consistency with the historical re-analysis datasets obtained from Open Meteo.

3.2 Data Processing

This section outlines how the data was processed after it was collected. The first subsection describes the data cleaning process for the SCADA data and then how a local upstream global wind direction and speed was calculated. Additionally, it presents how the data was split and normalized before training. Finally, it explains how the SCADA data was upsampled.

3.2.1 Data Cleaning

High-quality input data is crucial for model performance, particularly when working with SCADA data, which is often noisy and incomplete. This section outlines the preprocessing steps applied to determine turbine operational status, filter invalid data points, and handle missing values in the SCADA data. The result from each cleaning step is shown in section A.2 in Appendix A.

3.2.1.1 Turbine Operation Status

A feature indicating whether each turbine was operational was added in the SCADA data. A turbine was classified as inactive if its power output was zero or negative, with a lower threshold of -100 kW applied to account for periods during which the turbine consumed auxiliary power without producing electricity. In such cases, the

recorded power output was adjusted to zero to represent zero power generation. Power output readings below -100 kW were regarded as anomalous and excluded from the analysis by assigning them a value of NaN. However, such extreme outliers did not occur in the dataset.

For model training purposes, only data instances in which at least eight turbines were operational were retained. This filtering criterion reflects typical wind farm operations, as it is rare for most turbines to be offline simultaneously. Moreover, such instances offer limited informational value for predictive modelling.

3.2.1.2 Removing Curtailed Turbines

To filter out data affected by curtailment, power output values were set to NaN for any turbine operating in a curtailment mode other than mode 0 or mode 18. Although mode 18 in the provided SCADA data technically indicates operation under an external set point it is often set even during normal operation due to limitations in the turbine control software. As a result, mode 18 alone is not a reliable indicator of curtailment and was not excluded.

Since curtailment events are not known in advance by farm operators, data from affected turbines cannot be used for forecasting. Removing these values helps to reduce noise in the dataset, as curtailment modes imply that a turbine is not producing power consistent with the inflow conditions.

3.2.1.3 Removing Outliers

To further clean the dataset, outliers that deviated significantly from the theoretical power curve were set to NaN. Outliers were identified by comparing the recorded power output to the expected output in the power curve given the wind speed, and data points falling far outside acceptable error margins were discarded. The aim was to filter out data points that are likely due to extreme operational anomalies or data recording errors, rather than typical wake-induced variations.

3.2.1.4 Handling Missing Data

To ensure consistency of the dataset, all rows containing missing power output values or NaN entries for any turbine were removed. This was done given the significance of turbine interactions in wind farm modelling. Retaining incomplete observations where any of the turbines operational status and power output is not known could lead to inconsistencies, ultimately impacting both training and prediction accuracy.

3.2.2 Data Splitting and Sampling

Before training the ML models the dataset was split into 80% for training, 10% for validation, and 10% for testing. Since the data is a time series, the split was performed by randomly selecting a starting point and extracting the subsequent 24 hours of data, while ensuring that each data point appeared in only one of the three sets. This approach helps to prevent the validation and test sets from being too similar to the training data, which in turn reduces the risk of data leakage.

3.2.3 Normalization

All input and output features were normalized using min-max scaling to the range $[0, 1]$, based on statistics from the training data. For features with known physical bounds, such as the sine and cosine components of cyclical variables, the predefined limits were used directly instead of computing them from the data. Normalizing the features ensures that all features operate on a comparable scale, which helps to stabilize and to speed up the training of the ML models.

3.2.4 Construction of Upstream Global Wind Data

To improve the estimates of global wind speed and wind direction, an upstream global wind (UGW) dataset was constructed from the SCADA data which records wind speed and wind direction behind each turbine. Since the IFS re-analysis data has a low spatial resolution, does not account for local topography, and is not based on direct measurements at the site, it can be insufficient for high-precision modelling.

The upstream turbines are least affected by wake effects, as they are exposed to the undisturbed incoming wind, and therefore provide measurements closest to the true global wind conditions. To simplify the selection of upstream turbines, the artificial global wind speed and direction were calculated as the average from the two turbines with the highest recorded wind speeds in the SCADA data at each time step.

North calibration was performed before calculating the upstream global wind speed and direction to ensure consistency across turbines. North calibration adjusts each turbine's recorded wind direction to a common true north reference, correcting for installation biases or sensor misalignments, and was carried out using FLORIS-based Analysis for SCADA data (FLASC) [37]. FLASC is an open-source tool designed to correct wind direction measurements across wind farms by aligning them based on wake interactions. This step was essential to align wind direction measurements across the farm before averaging them to estimate the global wind direction. However, FLASC compares energy ratios computed from SCADA data with those simulated by FLORIS based on the farm layout. Since FLORIS is a steady-state simulation tool that simplifies wake dynamics, this introduces potential inaccuracies in the north calibration.

3.2.5 Construction of Hourly Data

The resolution of the provided SCADA data is 10 minutes. To potentially reduce noise and improve model training and performance, the data was also upsampled to a 1-hour resolution. It was done by averaging each feature using the data points within a 30-minute window before and after each full hour.

The upsampling was conducted before filtering and the upsampled data was filtered according to the same process as the 10-minute data. Only data where each turbine's curtailment mode remained consistently 0 or 18 throughout the hour were retained. Furthermore, only instances where each turbine were operational or not for the entire hour were included. The dataset was then split according to the same index as for the 10 minutes resolution SCADA data. Since the splitting was initial-

ized by randomly selecting a timestamp, upsampled hours with overlapping indices in the training, validation and test datasets were removed for simplicity.

3.3 Graph Representation of the Wind Farm

The following section describes how the wind farm is represented as a graph, including the node features and edge attributes used, as well as how they were calculated.

3.3.1 Graph Connections

Each row (i) or time step in the dataset is represented as a graph, capturing the state of the wind farm. Turbines are modelled as nodes within the graph, and the connections between them are determined using a simplified wake model.

The wake model used is shown in Figure 3.1, where D denotes the diameter of the wake and α the wake angle. Together with the wake length these parameters determine the expansion of the wake. The wind direction (wd) is assumed to be normal to the rotor plane, corresponding to the condition where the turbine nacelle is aligned perpendicularly to the incoming wind. In Figure 3.1, a downstream turbine is within the wake of the upstream turbine if it is located within the shaded region.

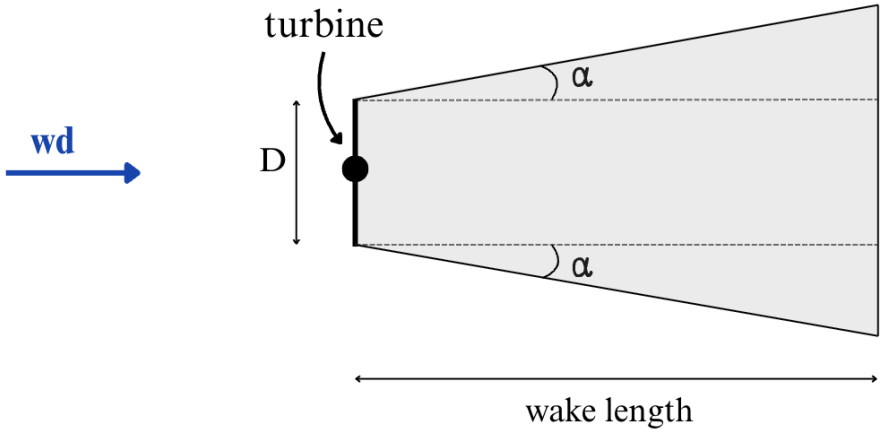

Figure 3.1: Illustration of the wake model for a single turbine. This model was used to determine which turbines are in wake of each other. D is the wake diameter and α is the wake angle.

When constructing the graphs a fixed value for D was used, corresponding to twice the rotor diameter of the wind turbines. The wake angle was varied across the following values: 3°, 10°, 20°, 40°, 70°, and 85°, along with a fully connected graph. A wake length of 2720 m was used, corresponding to 20 rotor diameters.

Figure 3.2 provides an overview of the process used to create the graphs, which consists of the following steps:

1. For each turbine, a wake region was generated based on the wind direction (wd) at time t, using the predefined wake model.

2. Directed edges were created from upstream turbines to downstream turbines that fall within their wake regions.
3. The clockwise relative angle ($\theta_{v,u}$) between the vector defined by the directed edge from node u to node v and the wind direction was computed for time t. This angle, $\theta_{v,u}$ was included as an edge attribute, along with other features. Node attributes were also added to the graph.

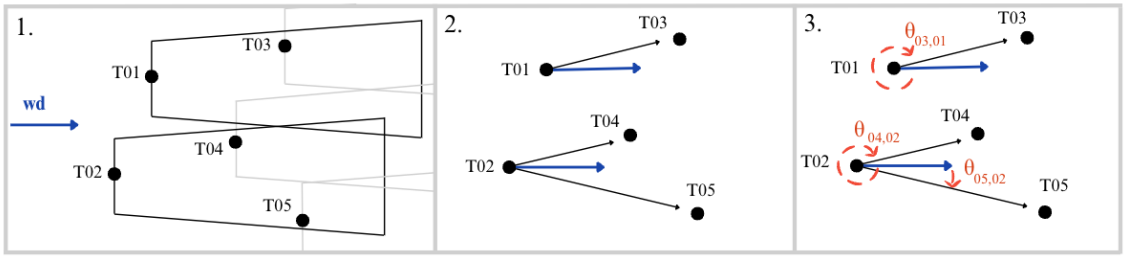

Figure 3.2: Steps used to build the graphs. (1) Wake areas were computed for each turbine based on the wind direction (wd). (2) Connections (edges) were added to the graph to represent wake interactions. (3) The clockwise angle $\theta_{v,u}$ between the wind direction and the edge (v, u) was calculated. This angle, $\theta_{v,u}$ was included as an edge attribute, along with other features. Node attributes were also added to the graph.

3.3.2 Node Features and Edge Attributes

In the graph representation, each node includes global features G such as wind speed, wind direction, other weather variables, hour of the day, and month of the year. The global features are identical across all nodes, reflecting conditions shared throughout the wind farm.

Wind direction was encoded using its sine and cosine components to preserve the continuity between 0° and 360°, and the same transformation was applied to the time features hour of the day and month of the year. Additionally, the turbine operation status t_v for each turbine was included as a local feature to signal which turbines were operational.

The edge attributes were defined by computing the Euclidean distance between turbines in the horizontal plane ($d_{xy,vu}$), the absolute difference in hub height, due to elevation variations across the wind farm site ($d_{z,vu}$), and the clockwise angle between the edge vector and the wind direction (θ_{vu}), shown in step 3 in Figure 3.2. This angle was transformed using its sine and cosine components ($\theta_{sin,vu}$,$\theta_{cos,vu}$).

3.4 Baseline Models

To evaluate the performance of the developed ML models, two baseline models were used. Both baselines were tested using two sources of global weather conditions: the IFS re-analysis data and the UGW dataset. The predicted power outputs were then compared against the historical power measurements in the SCADA data on the validation set.

3.4.1 Power Curve

The first baseline model is based on the wind turbine's theoretical power curve in the farm, which defines the expected power output for a given wind speed. The global wind speed was mapped to the power curve to predict the theoretical power output for each turbine. Since the power curve is identical for all turbines it does not account for turbine interactions or wake effects within the farm. This baseline provides a reference for how well simple theoretical model aligns with real-world turbine behaviour.

3.4.2 FLORIS

The second baseline uses the FLORIS simulation framework to estimate power output for the given wind farm layout. FLORIS includes simplified modelling of wake effects, allowing it to capture reductions in power generation for downstream turbines. In this study, the Gaussian wake model implemented in FLORIS was chosen, as it allows for the specification of turbine operational status (i.e., on or off).

Turbulence intensity (TI), which influences wake recovery and downstream turbine performance, is incorporated in the FLORIS model as a global input variable, along with wind speed and direction. The TI was integrated into the simulation framework through two distinct methods:

- Constant TI Approach: TI was fixed to a value of 10%, which was uniformly applied across all simulations.
- Variable TI Approach: TI was estimated as a function of wind speed using the Normal Turbulence Model (NTM) [38]. The standard deviation of the wind speed fluctuations, denoted σ, was calculated as:

$$\sigma = I_{\text{ref}}(0.75 \cdot \bar{v}_{ws} + c) \tag{3.5}$$

  where $I_{\text{ref}} = 0.14$ denotes the reference turbulence intensity, $c = 0.75$ is a constant, and $\bar{v}_{ws}$ is the mean wind speed at hub height over a given time period. The TI was then computed according to its definition in Equation 2.1. To estimate a representative value of I_{ref} from the UGW dataset, only wind speeds in the range of $15\,m/s \pm 0.2\,m/s$ were used. The standard deviation of these wind speeds was inserted into Equation 3.5. Finally, the 90th quantile of the obtained I_{ref} values were selected as the reference value.

The FLORIS baseline reflects how well a simplified physics-based model performs in power prediction.

3.5 Traditional Machine Learning models

The traditional ML methods used to predict power output for each turbine are RF, XGBoost, kNN regressor and MLP. This section outlines the features provided as inputs for the models and the strategy used when training them.

3.5.1 Input Feature Representation

Different sets of input features were evaluated for all these models. The simplest configuration included wind speed, wind direction, and turbine operational status. The traditional ML models operate on tabular input data and do not inherently account for spatial dependencies between turbines. Therefore, the in-degree of each node in the graph was calculated and added as a feature. It was incorporated to capture the wake effects of the farm and various wake angles were evaluated. The environmental and temporal variables temperature, hour of the day, and month of the year were also considered.

3.5.2 Training Configuration

This section details the methods used when training the traditional ML methods, including what datasets were used and how the hyperparameters were tuned.

3.5.2.1 Training Strategy

The RF, kNN, and XGBoost models were trained using two different strategies. In the first approach, a separate model was trained for each turbine using only global and turbine-specific input features. In the second approach, a single model was trained for all turbines collectively, using both global and local features for all turbines. This approach was motivated by the correlation between turbine outputs, where leveraging information from other turbines may improve prediction performance. However, both the scikit-learn implementation of RF [39] and the XGBoost implementation [40] used has not been found to support a joint loss function. Instead, the loss for each target is treated independently.

The MLP model was only trained using the second approach with one output target corresponding to each turbine. The training loss function utilized for the MLP was Mean Squared Error (MSE). To prevent the loss from being skewed by inactive turbines, observations corresponding to non-operational turbines (defined as those with zero power output) were excluded from the loss computation through masking. This was done as the power output from these are known to be zero.

All the traditional ML models were trained using the IFS re-analysis data on a 10-minute resolution and using both 10-minute and 1-hour resolution UGW data. Since the target variables were also upsampled in the 1-hour resolution dataset each model was evaluated on both resolutions of the UGW dataset to enable a fair comparison.

3.5.2.2 Hyperparameter Tuning

The RF, XGBoost and kNN models were hyperparameter tuned using grid search, a method that systematically evaluates all combinations of a predefined set of hyperparameter values to find the best-performing configuration. For the kNN model, the Euclidean distance was used as the distance metric, and the number of neighbours (k) was tuned. In the case of RF, the number of estimators, minimum sample split, and maximum tree depth were tuned. Similarly, for XGBoost, the tuned hyperparameters included the maximum depth, learning rate, and number of estimators.

Hyperparameter tuning for the MLP was performed using Optuna [41], a framework for automated parameter optimization. A total of 10 trials were conducted, tuning the number of hidden layers, number of neurons per layer, batch size, and learning rate. Additionally, early stopping was used to prevent the MLP models from overfitting to the training data and to reduce computational cost during hyperparameter tuning.

3.6 Graph Neural Network

This section presents the GNN model created for wind power prediction. Unlike conventional ML approaches, which treat turbines as independent units, the GNN captures spatial dependencies within a wind farm by representing turbines as nodes in a graph. The graph's edges may represent wake interactions and other spatial correlations, enabling the model to more effectively learn the complex inter-turbine dynamics that influence power output.

3.6.1 Architecture

For the GNN architecture an Encoder-Processor-Decoder structure was used , see Figure 3.3. This allows the message passing in the processor to occur in a higher dimensional space, which improves the expressivity of the GNN [42].

3.6.1.1 Encoder

The encoder converts raw node and edge features, $\mathbf{x}_v$ and $\mathbf{e}_{vu}$ respectively, into latent representations that are suitable for processing by the GNN. It consists of two distinct MLPs: one dedicated to node feature encoding and the other to edge feature encoding. Each MLP is composed of a sequence of fully connected layers, combined with ReLU activation functions and layer normalization. These MLPs project the input features into a higher-dimensional latent space, enabling more expressive and informative representations. The output of the encoder includes the transformed node features $\mathbf{h}_v^{(0)}$ and edge features $\mathbf{a}_{vu}$, which are subsequently passed to the message-passing processor block in the GNN.

3.6.1.2 Processor

The processor enables information propagation across the graph through a stack of message-passing layers based on GENConv operations [35]. Each GENConv layer updates node representations by aggregating feature information from neighbouring nodes and their associated edges. Feature aggregation is performed using additive operations over the incoming connections, followed by an MLP that includes two fully connected layers, each followed by layer normalization and ReLU activation.

3.6.1.3 Decoder

The decoder maps the final node embeddings produced by the processor to task-specific prediction outputs $\hat{y_v}$ for every node v. It employs an MLP with ReLU

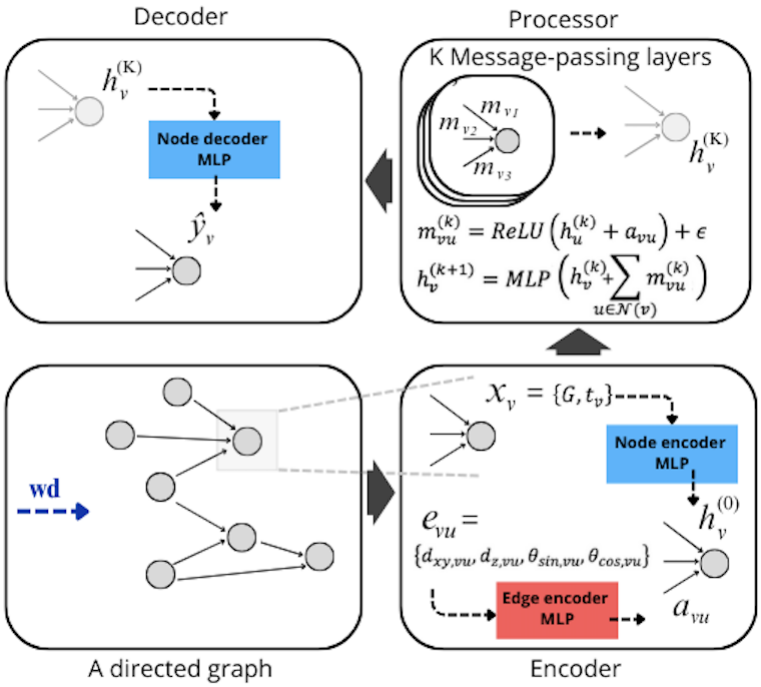

Figure 3.3: Overview of the GNN architecture. The wind farm is modelled as a directed graph based on wind direction (wd). Each node (turbine) is encoded using a node encoder MLP that processes local features $\mathbf{x}_v = \{G, t_v\}$, where t_v denotes the turbine's operational status and G the global features. Edge features $\mathbf{e}_{vu}$ include information such as horizontal and vertical distance, and relative angles. These are processed by an edge encoder MLP. The encoded graph is passed through K message-passing layers where node representations $\mathbf{h}_v^{(k)}$ are updated using aggregated messages from neighbouring turbines. Finally, the decoder MLP predicts the output $\hat{y}_v$ for each node v.

activations that processes the node embeddings and projects them into the desired output space. This block generates the final predictions for each node, such as predicting the power output of individual turbines. In multi-task learning settings, the decoder can produce multiple output values per node, each corresponding to a distinct target task.

3.6.2 Training Configuration

This section details the methods used when training the GNN, including which datasets were used and how the hyperparameters were tuned.

3.6.2.1 Training Strategy

Similarly to the traditional ML models, various input feature sets were systematically evaluated to assess their impact on predictive performance. Once the optimal feature configuration was identified, further analysis was conducted to examine how the wake angle used to define graph connectivity influences model performance. Sub-

sequently, a multi-task learning approach was explored using the best-performing graph configuration. In multi-task learning, a single model is trained to predict multiple related outputs simultaneously, which can facilitate shared representation learning and improve generalization, particularly when tasks exhibit strong interdependencies [43]. For wind power forecasting, auxiliary tasks such as the prediction of turbine-level SCADA measurements (e.g., local wind speed, wind direction, and nacelle orientation) were incorporated alongside power output. These additional supervisory signals can help the GNN learn more robust representations leading to improved power output predictions.

3.6.2.2 Training Settings

The GNN was trained using the MSE as the loss function, computed across all operational turbines. Similarly to the MLP, turbines that are inactive during specific time steps were excluded from the loss computation, as their zero power output is often known in advance by the farm operator and does not provide a meaningful learning signal.

Optimization was conducted using the Adaptive Moment Estimation (Adam) optimizer, which adaptively modifies individual learning rates for each parameter based on the first and second moments of gradients. A cosine annealing learning rate scheduler was employed to progressively reduce the learning rate according to a cosine decay schedule to improve performance during the later stages of training. Early stopping was implemented based on validation performance to prevent overfitting and enhance generalizability.

3.6.2.3 Hyperparameter Tuning

Hyperparameter optimization was conducted on the best-performing GNN architecture using the Optuna framework [41]. A total of 10 optimization trials were executed, targeting key hyperparameters including the initial learning rate, batch size, dimensionality of the latent node embeddings, and the number of message-passing layers (implemented as GENConv layers). This tuning process aimed to maximize validation performance.

3.7 Evaluation Metrics

Multiple evaluation metrics were used to assess model performance, with turbines that were turned off excluded (masked out) in all cases. All metrics are given in kW and the maximum that a turbine can produce is 3800 kW.

The Root Mean Squared Error (RMSE) was calculated between the model-predicted and actual power outputs from the SCADA data. For each row in the dataset i, the Mean Squared Error (MSE) was first averaged across all M turbines. The final RMSE was then obtained by averaging over the the whole dataset of size N and taking the square root:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{M}\sum_{ti=1}^{M}(y_{i,ti} - \hat{y}_{i,ti})^2\right)} \tag{3.6}$$

The Mean Absolute Error (MAE) was computed by averaging the absolute error per turbine and then averaging over all time steps:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{M}\sum_{ti=1}^{M}|y_{i,ti} - \hat{y}_{i,ti}|\right) \tag{3.7}$$

RMSE is a common evaluation metric for wind power prediction as it is more sensitive to deviating data points [5] which can be important for creating a good predictive model. MAE on the other hand measures the average magnitude of prediction errors, meaning it is less affected by outliers, which can be beneficial to prevent the evaluation to excessively penalize the model for outliers. Therefore, both RMSE and MAE were used together to provide a more comprehensive assessment of model performance, capturing both the overall error magnitude and the potential influence of extreme prediction deviations.

To get an understanding of the performance at farm level mean absolute percentage error (MAPE) is used. It is calculated as:

$$\text{Farm MAPE} = 100 \cdot \frac{1}{N}\sum_{i=1}^{N}\left(\left|\frac{\sum_{ti=1}^{M} y_{i,ti} - \sum_{ti=1}^{M}\hat{y}_{i,ti}}{\sum_{ti=1}^{M} y_{i,ti}}\right|\right) \tag{3.8}$$

where the total prediction $\hat{y}_{i,ti}$ and power output $y_{i,ti}$ are summed. Then divided by the total power output. The absolute percentage error are then summed and divided by the size of the dataset N. MAPE provides the error as a percentage of the actual value, helping to understand how far the model's predictions are from the true values.

3.8 Testing on Forecast Data

To assess the generalization capability of the models for wind power forecasting, evaluation was conducted using historical forecast data. Forecast hours corresponding to the timestamps in the test set were extracted, and model performance was compared across both the test and forecast datasets.

This evaluation was conducted on models trained using both the IFS and the UGW dataset. Only the best-performing models were selected for this analysis. The models considered for evaluation was a subset of the models, as the meteorological variables extracted from the forecast data was limited to wind speed, wind direction and temperature. To benchmark the models' performance and generalizability, the wind farms' theoretical power curve was used as a baseline in the power forecasting task.

4

Results

This chapter presents the result of the evaluated models for wind power prediction in a wind farm. The analysis begins by establishing baseline performance using the wind farm's power curve, alongside simulations generated using the FLORIS model. These benchmarks serve as reference for evaluating the performance of the ML models.

The models are trained and evaluated on IFS re-analysis data and a manually constructed upstream turbine dataset (UGW). A systematic investigation is conducted to examine the influence of different input feature sets and graph configurations on model performance, considering both traditional ML models and GNN architecture. In addition, the study explores the integration of multi-task learning within the GNN framework to evaluate its effect on power prediction.

The result also includes a comparison of model performance when trained on datasets with 10-minute and 1-hour resolution. This analysis explores the sensitivity of predictive performance to the temporal resolution of the input data. Finally, the generalization capacity of the models is assessed using forecast data to evaluate their practical applicability in real-world wind power forecasting.

4.1 Baseline Models

As a baseline for evaluating model performance, comparisons are made against both the wind farm's power curve and FLORIS simulations. The results using the IFS re-analysis data are presented in Table 4.1, while the results obtained using the UGW dataset are provided in Table 4.2.

Baseline Model	RMSE (kW)	MAE (kW)
Power Curve	602	419
FLORIS (TI from NTM)	1180	841

Table 4.1: Results from the baseline models using IFS input. RMSE and MAE values are given in kW.

Baseline Model	RMSE (kW)	MAE (kW)
Power Curve	772	578
FLORIS (TI = 10%)	616	432
FLORIS (TI from NTM)	611	432

Table 4.2: Results from the baseline models using UGW input. RMSE and MAE values are given in kW.

4.2 Models Trained on IFS Re-analysis Data

The best-performing configurations of each traditional ML model and the GNN, trained on IFS data using wind speed, wind direction, and turbine operational status as features, are presented in Table 4.3. The results indicate that the models perform similarly and farm MAPE show prediction errors are above 50%. Meaning predictions deviate from the actual values by more than half of the actual values themselves and all models have similarly poor predictive performance.

Model	RMSE	MAE (kW)	Farm MAPE (%)
GNN	500	358	67
kNN	511	359	73
RF	514	360	62
XGBoost	509	356	62
MLP	532	368	60

Table 4.3: Best traditional ML models trained on IFS re-analysis data showing the performance metrics. RMSE and MAE values are given in kW.

4.3 Models Trained on UGW Data

This section presents the results from the models trained on the UGW dataset, generated from upstream turbines in the SCADA data to represent global weather data. It includes results for both traditional ML models and the GNN at a 10-minute resolution.

4.3.1 Traditional Machine Learning Models

Figure 4.1 presents the results of training individual models that predicts the power output of each wind turbine using various combinations of input features. All models are trained using wind speed, wind direction, and turbine operational status as baseline features. The labels in the figure indicate which additional features are included during training and the time feature set refers to both the month of the year and the hour of the day. The kNN regressor displays reduced performance with the inclusion of additional features, suggesting that it struggles to handle high-dimensional input data effectively. In contrast, both the RF and XGBoost regressors maintains consistent performance across different feature sets.

Figure 4.2 illustrates the results of training a single, multi-output model that predicts the power output of all turbines simultaneously. These models uses the same

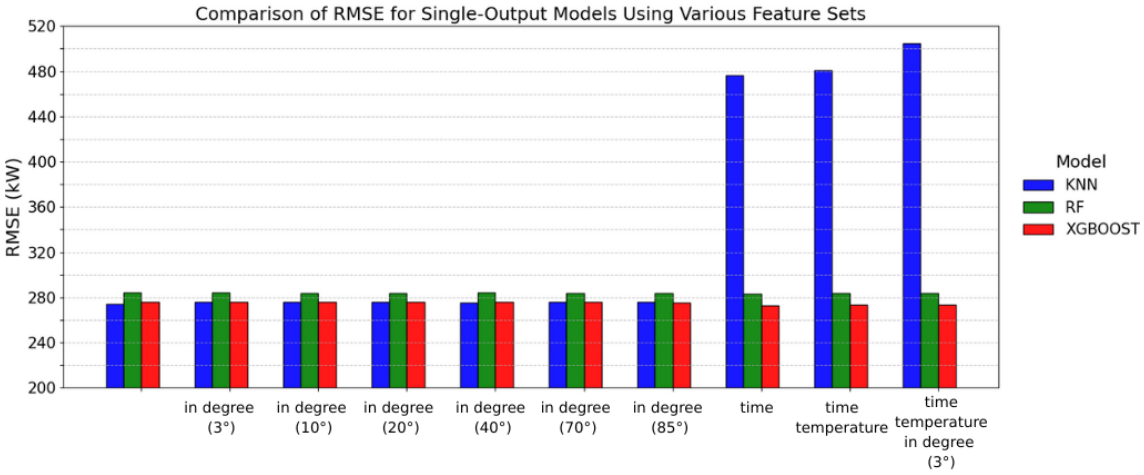

Figure 4.1: Models trained using one model per turbine. RMSE in kW for different sets of features.

baseline features, wind speed, wind direction, and turbine operational status, with the additional features specified in the figure. Among the evaluated models, the MLP achieves the highest performance when incorporating the in-degree feature, which is derived from a wake interaction graph constructed using a 20° wake angle. In contrast, the kNN model performs worse in this multi-output setting across all feature sets. Similarly, both the RF and XGBoost models exhibits increased prediction error when using a single multi-output model.

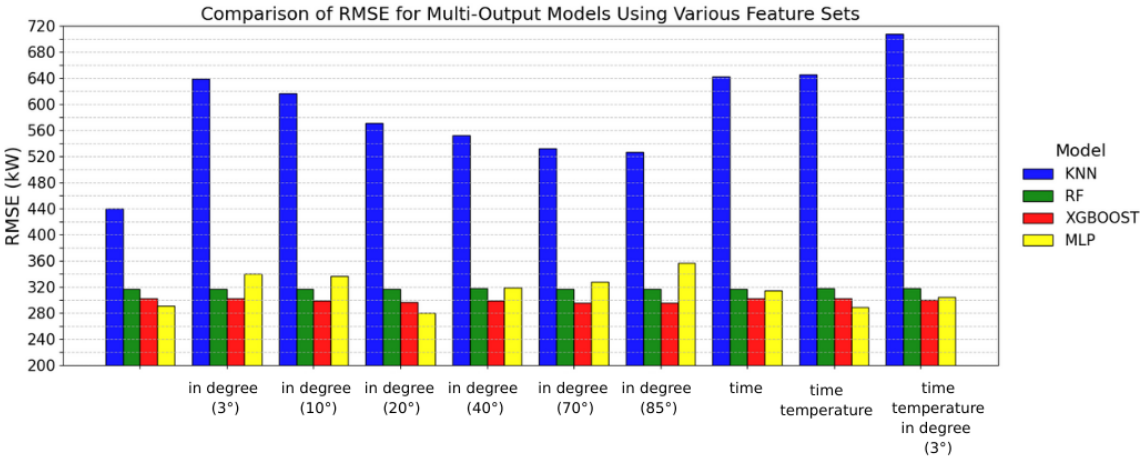

Figure 4.2: Models trained using one model for all turbines. RMSE in kW for different sets of features.

Table 4.4 summarizes the evaluation metrics for the best-performing models, along with their corresponding optimal feature sets. With the exception of the MLP, the most accurate results are obtained by training separate models for each turbine. For the kNN model, the optimal configuration includes only the baseline features (wind speed, wind direction, and operational status). In the case of RF and XGBoost, adding the temporal variables (month and hour) results in enhanced predictive performance. The predictive performance when training on the UGW data ranges

between 13% and 19%, depending on the model, meaning that the farm-level predictions deviate from the true values by only 13–19% on average.

Model	Additional features	RMSE (kW)	MAE (kW)	Farm MAPE (%)
kNN		274	183	16
RF	time	283	189	13
XGBoost	time	273	182	14
MLP	wa (20°)	281	188	19

Table 4.4: Best models trained on UGW data showing the performance metrics. RMSE and MAE values are given in kW and Farm MAPE in %.

4.3.2 Graph Neural Network

The GNN training is conducted in multiple stages, each aiming to reduce prediction error. This section presents the results from experimenting with different input feature sets, varying graph connectivities, and incorporating multi-task learning to assess its impact on performance.

4.3.2.1 Feature Exploration

The results in Table 4.5 show the impact of different input feature sets on the GNN model performance. All models are trained with wind speed, wind direction and turbine operational status as inputs. The features in the table shows which additional features are added in each model. In the second model, hour of the day and month of the year are added and in the third one, air temperature and relative humidity from the IFS re-analysis dataset are added as well. For all runs, the wake angle is fixed at 20 degrees. The results indicate that the RMSE does not significantly improve when adding time of day and month of year. Overall, all evaluation metrics show that the model performances are very similar across the different feature sets.

Additional Features	RMSE (kW)	MAE (kW)	Farm MAPE (%)
-	282	194	15
hour of day, month of year	281	194	15
hour of day, month of year, temperature, relative humidity	281	192	14

Table 4.5: GNN results with different input features, showing the performance metrics. All RMSE and MAE values are given in kW.

4.3.2.2 Graph Representation Exploration

To further explore how the connectivity of the graph can improve the result, different wake angle settings are evaluated with the best model from the feature comparison. Consequently, the input features for this model includes turbine operational status, global wind speed and wind direction, hour of day, and month of year.

The results in Table 4.7 indicate that the performance is relatively consistent across different wake angles, with 70° and 85° yielding the lowest RMSE. A fully

connected graph configuration results in lower error compared to the 3° wake angle. As shown in Table 4.6, a significant percentage of nodes are isolated at wake angle 3°, meaning they lack both incoming and outgoing edges. However, when the wake angle is 70° or greater, no nodes are isolated.

Graph Statistics	T00	T01	T02	T03	T04	T05	T06	T07	T08	T09	T10	T11	T12	T13	T14	T15
in-degree:																
3°	0.2	0.2	0.3	0.3	0.3	0.3	0.3	0.3	0.1	0.3	0.3	0.3	0.4	0.2	0.2	0.2
10°	0.7	0.8	0.9	0.9	1.1	1.1	1.0	0.9	0.5	0.9	1.1	0.9	1.2	0.7	0.6	0.7
20°	1.3	1.6	1.9	1.7	2.2	2.3	2.1	1.7	1.0	1.9	2.1	1.9	2.3	1.4	1.2	1.4
40°	3.1	3.2	3.9	3.5	4.3	4.4	3.9	3.4	2.1	3.6	4.2	3.9	4.3	2.8	2.5	2.9
70°	5.9	5.7	6.6	5.7	6.9	7.0	6.1	5.4	4.1	6.1	6.6	6.7	6.8	4.9	4.2	4.9
85°	7.1	6.7	7.7	6.6	7.9	8.0	7.2	6.4	5.1	7.2	7.6	7.8	8.0	6.0	5.1	5.8
Isolated (%)																
3°	67	54	68	53	56	56	53	59	75	59	56	63	49	55	56	59
10°	31	9	28	19	30	26	16	29	51	11	17	24	7	9	26	25
20°	10	0	2	7	18	2	5	11	34	0	3	1	0	0	8	9
40°	0	0	0	0	1	0	0	0	11	0	0	0	0	0	0	0
70°	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
85°	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.6: Graph connectivity statistics derived from the training dataset for different wake angles. The table presents the average in-degree and percentage of isolated nodes for each turbine (T00-T15). Increasing the wake angle generally enhances connectivity, reducing the number of isolated nodes.

Wake Angle	RMSE (kW)	MAE (kW)	Farm MAPE (%)
Fully Connected	281	191	14
3°	287	197	14
10°	284	194	13
20°	279	191	14
40°	275	189	16
70°	273	187	14
85°	273	188	16

Table 4.7: GNN results with different wake angles, showing the performance metrics all in kW except for MAPE.

Different turbines in the farm exhibited varying prediction errors. Figure 4.3 illustrates the layout of the wind farm with the 16 turbines. The figure highlights that with a low wake angle, Turbine 8, located in the outer corner of the farm, has the highest RMSE, which aligns with its high percentage of isolation in the graph. In contrast, turbines situated in the middle of the farm generally exhibit lower RMSE values across all wake angles.

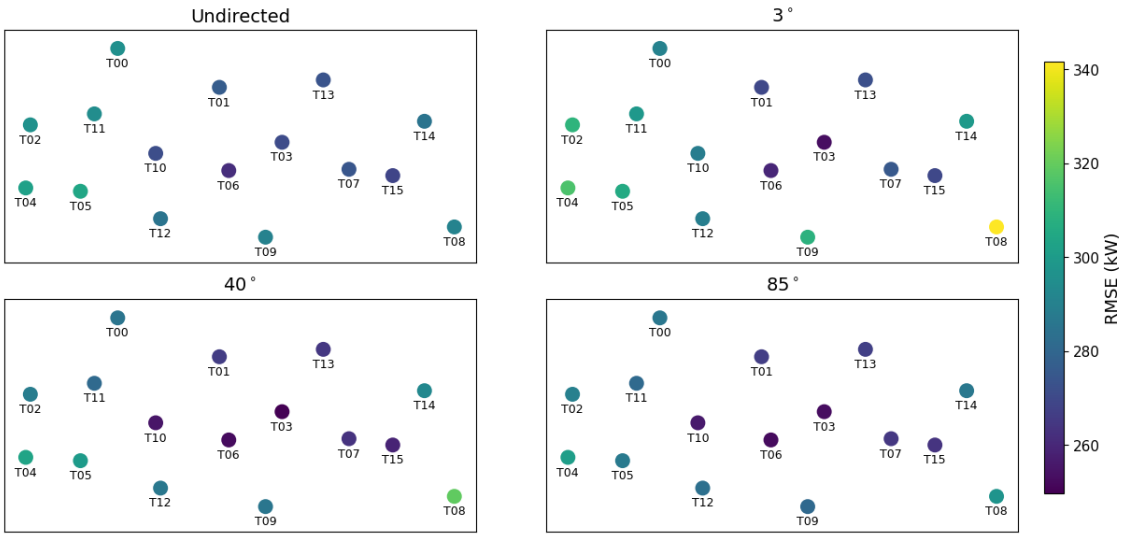

Figure 4.3: Visualization of RMSE per turbine for different graph configurations: fully connected graph, and graphs with wake angles of 3°, 40°, and 85°.

4.3.3 Multi-Task Learning

The impact of incorporating additional targets in the GNN is investigated to assess whether multi-task learning can enhance wind power prediction performance or not. Figure 4.4 presents the comparison between single-task and multi-task learning, illustrating the effect of adding local SCADA measurements as output targets for each turbine. Adding local wind speed at each turbine as an additional target shows some improvement, though the overall gain compared to single-task learning is minimal. Adding any other targets increases the prediction error for power generation.

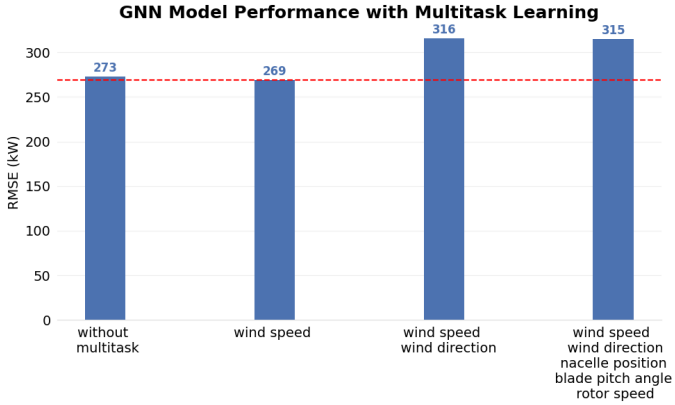

Figure 4.4: GNN results comparing single-task learning to multi-task learning with additional turbine-level SCADA measurements as output targets, displaying the RMSE.

4.4 Comparative Evaluation of Models

The best performing GNN and traditional ML models are shown alongside the baseline models evaluated on the UGW dataset in Figure 4.5. The top-performing GNN model is the multi-task model that includes wind speed as an additional output, while the XGBoost model trained separately for each turbine. The results in Figure 4.5 are displayed for turbine 3 and is the percentage distribution of absolute errors for each model. The average MAE for turbine 3 is shown as a dashed red vertical line.

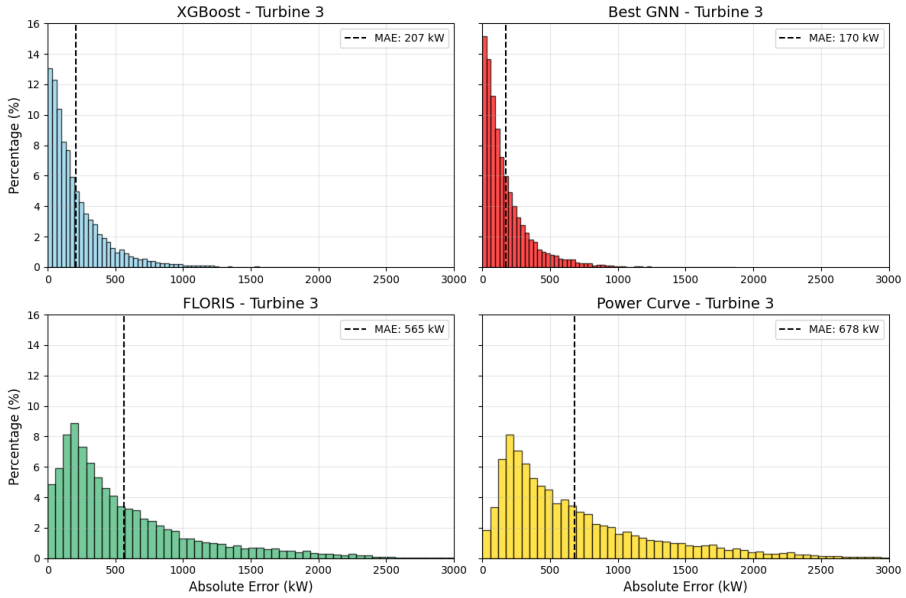

Figure 4.5: Absolute error distribution comparison for XGBoost, Best GNN, FLORIS, and Power Curve models for Turbine 3. The average MAE for turbine 3 is shown as a dashed black vertical line.

Figure 4.6 presents the predicted power output versus the target output for the best performing models. The data shown are the results from Turbine 3, which is situated centrally within the wind farm, making it representative of average operating conditions with wake effects from surrounding turbines. The figure shows how closely the models are able to predict the power output and offers insights into potential bias in predictions.

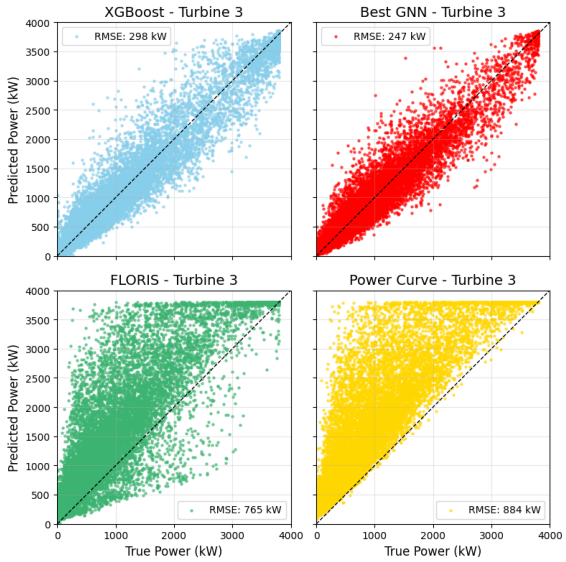

Figure 4.6: True vs. Predicted power comparison for XGBoost, Best GNN, FLORIS, and Power Curve models for Turbine 3.

4.4.1 Models Trained on 10-Minute Versus 1-Hour Resolution Data

The traditional ML models using features wind speed, wind direction, and turbine operational status are trained on both 1-hour and 10-minute resolution data to investigate whether decreasing the temporal resolution helps reduce noise and improve model performance. The results are presented in Table 4.8. "Models 10 min data" in the table are models trained on 10-minute resolution data, while those labelled "Model 1 h data" are trained on 1-hour resolution data. To ensure a fair comparison between the models, they are evaluated on both dataset resolutions regardless of their training resolution. The results show that the traditional ML models perform similarly when trained on the hourly and the 10 minutes resolution data. The best GNN model trained on the 10 min resolution SCADA data is included in the table as well. It is evaluated on both datasets, and all models perform better on the data which has an hourly resolution.

Model	Additional features	Model 10min data		Model 1h data	
		RMSE (10min, kW)	RMSE (1h, kW)	RMSE (10min, kW)	RMSE (1h, kW)
KNN	-	274	232	293	254
RF	-	284	248	288	240
XGBoost	-	276	248	279	230
MLP	-	291	250	286	237
GNN	time	273	241	-	-

Table 4.8: Comparison of traditional ML models trained on 1-hour and 10-minute resolution data. RMSE values in kW are reported for both resolutions to ensure fair comparison. The best-performing GNN model is included as a reference.

4.5 Predicting on Forecast Data

To get an understanding of the models ability to generalize to forecast data the best performing ML models, XGBoost and GNN, are evaluated on both forecasted weather data and the corresponding test data upsampled to 1 hour resolution. The results are shown in Figure 4.7. The x-axis indicates the time difference between when the forecast is made and the target time which is called lead time. The test data is seen as having a forecast lead time of zero hours, and thus lead time only applies to the forecast data.

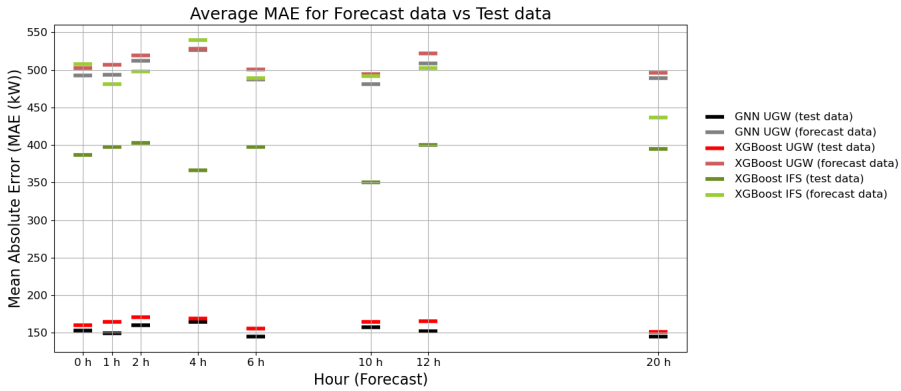

Figure 4.7: Comparison of models evaluated on forecast data and the re-analysis test data. The x-axis shows the forecast lead times for the forecast data and y-axis the MAE in kW.

The results show no clear trend with respect to forecast lead time. Models trained on IFS data and UGW data demonstrate comparable performance on forecast data. All models perform better when evaluated on their respective test datasets.

Table 4.9 provides a summary of the average MAE and RMSE values for each model, calculated across all forecast lead times. In addition to the best-performing ML models, the table includes results from evaluating the power curve on forecast data, which achieves the highest overall performance.

Model	Train data	MAE (kW) Test data	MAE (kW) Forecast data	RMSE (kW) Test data	RMSE (kW) Forecast data
XGBoost	IFS	376	483	551	701
XGBoost	UGW	158	506	270	740
GNN	UGW	148	498	247	731
Power curve		-	434	-	658

Table 4.9: Summary of MAE and RMSE for each model evaluated on both forecast data and re-analysis data from the test set. The metrics are averaged over all forecast lead times. The table includes results for the best-performing ML models (GNN, XGBoost trained on UGW and IFS data), as well as a baseline evaluation using the power curve, which outperforms the other models in terms of both MAE and RMSE when evaluated on the forecast data.

5

Discussion

This chapter presents an analysis of the results from the baseline models, traditional ML models and GNN used for wind power prediction. The discussion is structured around several key aspects. First, the performance of the baseline models is compared across datasets, with a focus on the limitations of simplified physical assumptions. Next, the traditional ML models results are discussed, particularly their response to input features and multi-output learning. Then, by exploring different wake angles, it is evaluated how varying degrees of graph connectivity affect prediction accuracy.

In addition, the effect of temporal resolution on model performance is analysed, followed by a discussion on how well the models generalize to forecasting wind power production. Finally, the reliability of the global wind speed and wind direction derived from SCADA data is discussed, along with its implications for model performance and potential improvements.

5.1 Models Performance Comparison

The ML models successfully perform better than the baseline models when trained and evaluated on both the IFS and UGW datasets. This improvement shows the models' ability to capture complex, non-linear relationships between global conditions and wind power output. In this section, the results are interpreted in greater depth, with a focus on model performance, input features, generalization capability, and limitations.

5.1.1 Baseline Models

Across the two datasets, IFS re-analysis and UGW, none of the baseline models consistently performs best. The UGW dataset is biased toward higher than average wind speeds across the farm, as it uses the average wind speed measured at the two turbines with the highest wind speeds (see Appendix A, Figure A.5). These are typically upstream turbines that are less affected by wake. Since the power curve does not account for wake effects, this biased wind speed likely contributes to the poorer performance of the power curve model on the UGW dataset compared to the IFS dataset. This is illustrated in Figure 4.6, which shows that the power predicted using the power curve and UGW data clearly overestimates the true power for a turbine located in the middle of the farm. In contrast, the IFS dataset has a lower spatial resolution, resulting in a smoother wind speed with less fluctuations that

may better represent the farm wide average wind speed.

The FLORIS model, which accounts for wake effects, performs better on the UGW data compared to the IFS data. This may be attributed to the UGW data more accurately representing the global wind conditions required as input to the FLORIS model. However, the inclusion of a calculated turbulence intensity for the farm did not result in a significant improvement in the model's performance. This is likely due to the turbulence intensity being derived directly from the UGW dataset. If the UGW data does not correctly represent the global wind conditions or contains inherent biases, the calculated turbulence intensity is unlikely to provide additional meaningful information to the model.

Moreover, neither model has been optimized or tuned to the input data. Consequently, the full potential of the baseline models has not been explored, and possible strategies for improving their accuracy have not been considered. For instance, the power curve model could be adapted to correct for known biases in the input data, such as systematic overestimation of wind speed. Similarly, the FLORIS model could be calibrated by adjusting its internal coefficients to better represent site-specific conditions and the characteristics of the available data, although this would involve a computationally expensive process.

5.1.2 Traditional Machine Learning Models

For the traditional ML models, all models trained on the UGW dataset outperforms those trained on the IFS re-analysis data. All models achieves the best performance when one model is trained per turbine, except for MLP which was only trained in a multi-output setting. Meaning it was only trained on predicting the power output for all 16 turbines simultaneously. In the multi-output setting, the operational status features are included as inputs, and it may be challenging for the models to identify which turbine each feature corresponds to. This limitation is especially pronounced for kNN, as it relies solely on the nearest neighbours to make a prediction and lacks an internal mechanism for explicitly associating features with specific turbines.

The inclusion of additional features beyond wind speed, wind direction, and turbine operational status does not affect the performance of the RF and XGBoost models. This suggests that the added temporal and environmental features do not contribute with meaningful information to these models. In contrast, the MLP and kNN regressors exhibits greater sensitivity to changes in the input features. For kNN, this is due to its reliance on the closest training samples in the feature space, making it highly responsive to changes in input dimensionality. Since, MLP is a parametric model it is possible that this makes it more sensitive to additional input features, particularly when these introduce noise or are redundant.

In the multi-output setting, training RF and XGBoost with one model for all turbines does not allow the use of a custom loss function that jointly considers the errors across all outputs. Instead, each output is treated independently, which may contribute to the reduced performance observed for these models. In contrast, the MLP is trained using a joint loss function across all turbines. In spite of that, XGBoost still performs slightly better then the MLP when trained using separate models for each turbine. One possible explanation is that the power output from

each turbine implicitly captures interactions between turbines, including wake effects. Since downstream turbines are affected by the reduced wind speed resulting from upstream turbines at a given wind direction, their power output inherently reflects these dependencies. This allows the individual models to learn some of the underlying wake dynamics, even without explicit spatial modelling.

5.1.3 Graph Neural Network

The best-performing GNN achieves performance comparable to that of the best traditional ML model, which is XGBoost. The inclusion of temporal features as input to the GNN results in only marginal improvements, likely attributable to randomness in the training process rather than meaningful contributions from the features themselves. Similar to the ML models, the environmental features does not provide valuable information to the GNN.

In Figure 4.3, it is evident that the turbines located in the middle of the farm achieve the highest accuracy in wind power prediction, despite being the most affected by wake effects. This shows that the GNN successfully captures wake effects in the wind farm. The higher RMSE for upstream turbines may be explained by the fact that upstream turbines are frequently adjusted under certain wind conditions to minimize their impact on downstream turbines and to optimize overall power production. Consequently, the power output in the SCADA data associated with upstream turbines may exhibit greater variability or uncertainty.

GNN performance varies slightly depending on the wake angle, which defines the connectivity of the graph. Performance improves up to a wake angle of 70°, at which point no graph contains isolated nodes. In Figure 4.3, it is observable that nodes frequently isolated exhibit higher RMSE compared to those with more connections. This observation, combined with the fact that a 3° wake angle, resulting in a high proportion of isolated nodes and a low average in-degree, as shown in Table 4.6, performed worse at the farm level than a fully connected graph, suggests that defining graph connectivity solely based on wake relationships may not be optimal. However, the limited predictive improvement with varying wake angles may be attributed to the small size and fixed layout of the wind farm, which could reduce the benefits of explicitly modelling spatial relationships through a graph structure. This could also explain why simpler ML models performed comparably well.

One potential improvement to the GNN involves revising how graph connectivity is defined. Rather than establishing connections solely based on a fixed wake angle threshold, all turbines located behind an upstream turbine within a 90° wake angle could be connected. A binary edge feature could then be introduced to indicate whether a downstream turbine is actually in the wake of another turbine, set to 1 if in wake and 0 otherwise. This approach would maintain higher graph connectivity, which is beneficial for GNN performance, while still capturing wake-specific interactions through edge-level information.

Furthermore, multi-task learning does not lead to a significant improvement in power prediction accuracy. Including the local wind speed at each turbine as an output results in a slight performance gain, whereas the addition of wind direction, nacelle position, and blade pitch angle for each turbine lead to a decline in

model performance. This may be due to the introduction of noise in the output features. For instance, wind direction is calibrated relative to north using a simplified method, which means the offset in the global wind direction is mixed with the individual offsets for each local turbine, potentially introducing noise. Another possible explanation is the limited diversity of feature combinations in the dataset. Specific configurations of blade pitch angle, nacelle position, and global wind conditions may occur too infrequently to provide consistent and informative patterns for the model to learn from.

5.2 Effect of Temporal Resolution

All models trained on both 10-minute and 1-hour resolution data exhibits similar performance when evaluated on 10-minute resolution data, with the exception of kNN, which shows a noticeable decline in performance. Reducing the temporal resolution decreases the number of available data points, which likely affects kNN negatively due to its reliance on having a sufficient number of neighbours. Since upsampling reduces variability in the dataset, the comparable performance of models trained on different temporal resolutions suggests that the high-resolution models are not fully capturing the underlying variability or complexity of the data. Instead, the models appear to learn general patterns, rather than demonstrating strong predictive capabilities. This may also explain why the inclusion of additional input features does not result in performance improvements, as the models might struggle to extract and utilize more intricate relationships within the data.

One contributing factor could be the limited amount of training data. Although the dataset spans a five-year period, only 59.8% of the data is kept after filtering. Consequently, the range of observed weather conditions is limited. Expanding the training dataset could improve the models' ability to learn more complex relationships, thereby enhancing their overall predictive performance.

5.3 Wind Power Forecasting

The models trained on UGW and IFS data perform similarly when evaluated on the historical forecast dataset. However, the power curve model yields the best performance, indicating that the trained models do not generalize well to the forecasting data. This is likely due to discrepancies between the training and forecasting datasets. One of the most prominent differences is in spatial and temporal resolution: UGW has higher resolution in both dimensions, while IFS has lower spatial resolution. Moreover, in the forecast data, wind variables are interpolated to hub height (117 m), whereas the IFS dataset provides wind data only at 100 m. In the UGW dataset, wind speed is adjusted using a transfer function, and wind direction is corrected through north calibration. These differences in data sources and preprocessing methods likely introduce inconsistencies that contribute to the reduced model performance on forecast data.

The predictive performance of the models does not exhibit any clear dependency on forecasting lead time. This is unexpected, as forecast uncertainty typically in-

creases with lead time, and a 0-hour forecast is generally very similar to the actual weather conditions. This lack of variation is most likely attributable to differences between the datasets used. Additionally, both the number of data points and the range of forecast lead times are limited, which may not be sufficient to reveal any clear dependency between lead time and predictive performance. It is also possible that the inherently stochastic nature of wind contributes to substantial forecast uncertainty even at shorter lead times.

One potential way to reduce the discrepancies between datasets is to train models directly on the MET Norway (metno) forecast data by first conducting a re-analysis on it. This approach would involve aggregating past 6 hours forecasts and correcting for systematic differences between them, thereby aligning the training and inference conditions. Such a strategy could improve model performance, as the same weather model and consistent spatiotemporal resolution would be used throughout. However, such data would still contain uncertainty in the weather conditions as it is based on a forecast data, which could result in noise in the weather data.

Furthermore, given the inherent uncertainty associated with weather-related data, the use of probabilistic models that can represent uncertainty in their predictions may be advantageous. For power forecasting tasks, alternative model architectures that can provide a range of likely outcomes rather than a single point estimate might be better suited. Incorporating information about forecast confidence or the underlying distribution could enhance both the robustness and interpretability of the predictions.

5.4 Global Wind Speed and Wind Direction

This section describes how global wind speed and wind direction are defined and used in the UGW dataset, along with the limitations and assumptions involved in their estimation. It outlines potential sources of bias in the underlying SCADA data, the implications of using a single global wind vector to represent farm-wide inflow conditions, and how these choices affect model performance and generalization.

5.4.1 Sources of Bias in Wind Measurements

The SCADA data used to derive global wind speed and wind direction in the UGW dataset contains inherent limitations and potential biases. North calibration is carried out using FLASC, which in turn relies on FLORIS. If this calibration is inaccurate, the resulting global wind direction estimates may be unreliable. In the UGW dataset, global wind direction is computed by averaging the wind directions of the two turbines with the highest wind speeds. However, if individual turbine measurements retain residual offsets from true north, and if these offsets differ between turbines, this process introduces noise into the aggregated wind direction. This issue is also relevant given that graph connectivity in the GNN model is defined based on wind direction.

Another source of bias arises from the transfer function used to correct wind measurements taken from behind the turbine rotor. If this correction function introduces varying biases across different turbines, it can further distort the estimation of global

wind conditions. Additionally, the exact formulation of the transfer function is proprietary and undisclosed by the original equipment manufacturer, who may modify it without notification. Consequently, the global upstream wind speed derived from corrected measurements may contain inconsistencies that negatively impact model performance.

5.4.2 Global Inflow Conditions

The developed models use a single, global wind speed and wind direction as input. The wind farm spans an area of approximately 1.5 km by 3.5 km (see Appendix A, Figure A.2). Wind variability occurs at multiple spatial scales and is critical to capture for accurate wind power prediction. However, the available re-analysis and forecast datasets found and used in this study provide limited, or no information about wind variability at the local (farm) scale. Therefore, the best available option is to train the models with a global wind speed and direction, which may limit their performance.

Wind conditions could vary significantly across the farm, meaning that a single wind measurement may not accurately represent the inflow conditions experienced by the upstream turbines (see Appendix A, Figure A.6). This highlights the potential need for incorporating the spatial variability of wind speed into the models, or for using weather data that more accurately reflects local inflow conditions at the turbine level. However, a key limitation is that current forecasts, particularly at higher altitudes, often lack the spatial resolution necessary to capture these localized inflow conditions. Probabilistic models may also be more suitable in this context, as they might be able to better handle uncertainty and variability of inflow conditions across the farm.

6
Conclusion

This thesis explored the use of traditional ML models such as XGBoost and kNN, as well as GNN capable of capturing spatial relationships within a wind farm, for wind power prediction. The results demonstrate the potential of ML models in wind power prediction as all models significantly outperform baseline methods. However, all models achieve similar performance and seem to only have limited predictive ability as additional weather variables do not improve performance. With the limiting factor most likely being the available historical weather conditions data and the amount of data available and used for training.

The study also highlights the importance of reducing the discrepancy between training and forecasting datasets in order to make wind power forecasting more reliable. Addressing this challenge, along with improving data quality and quantity, will be essential for advancing the accuracy and applicability of ML models in renewable energy forecasting. This, in turn, will ultimately enable greater integration of renewable energy into power systems.

Bibliography

[1] National Renewable Energy Laboratory, "Floris: A python library for wind farm layout optimization." `https://nrel.github.io/floris/`, 2025. Accessed: 2025-01-23.

[2] Open-Meteo, "Historical weather api documentation," 2025. Historical weather data provided by Open-Meteo.

[3] MET Norway, "Numerical weather prediction products," 2025. Documentation and data access for MET Norway's Numerical Weather Prediction products.

[4] Y. Wang, R. Zou, F. Liu, L. Zhang, and Q. Liu, "A review of wind speed and wind power forecasting with deep neural networks," *Applied Energy*, vol. 304, p. 117766, 2021.

[5] Z. Liu, H. Guo, Y. Zhang, and Z. Zuo, "A comprehensive review of wind power prediction based on machine learning: Models, applications, and challenges," *Energies*, vol. 18, no. 2, p. 350, 2025.

[6] D. Song, X. Tan, Q. Huang, L. Wang, M. Dong, J. Yang, and S. Evgeny, "Review of ai-based wind prediction within recent three years: 2021–2023," *Energies*, vol. 17, no. 6, p. 1270, 2024.

[7] S. Government, "Sweden's draft integrated national energy and climate plan," 2019. Accessed: 2024-03-19.

[8] S. W. E. Association, "Wind power statistics and forecast - quarter 4 2024," 2024. Accessed: 2024-03-19.

[9] S. E. Agency, "Scenarier över sveriges energisystem 2023," 2023. Accessed: 2024-03-19.

[10] S. Vindenergi, "Ppa - en nyckel i finansieringen av ny kraftproduktion." Accessed: 2024-04-01.

[11] Nordic Imbalance Settlement, "Nordic imbalance settlement handbook: Instructions and rules for market participants." Version 3.4, May 2022. For contractual basis.

[12] G. Duthé, F. de Nolasco Santos, I. Abdallah, P.-É. Réthoré, W. Weijtjens, E. Chatzi, and C. Devriendt, "Local flow and loads estimation on wake-affected wind turbines using graph neural networks and pywake," *Journal of Physics: Conference Series*, vol. 2505, no. 1, p. 012014, 2023.

[13] Y. Song, D. Tang, J. Yu, Z. Yu, and X. Li, "Short-term forecasting based on graph convolution networks and multiresolution convolution neural networks for wind power," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1691–1702, 2023.

[14] Y. Liu, K. Huang, J. Liu, P. Zhang, and Z. Liu, "Available power estimation of wind farms based on deep spatio-temporal neural networks," *Frontiers in Energy Research*, vol. 11, May 2023.

[15] E. A. Tuncar, Şafak Sağlam, and B. Oral, "A review of short-term wind power generation forecasting methods in recent technological trends," *Energy Reports*, vol. 10, pp. 197–209, 2024.

[16] S. Huang, C. Yan, and Y. Qu, "Deep learning model-transformer based wind power forecasting approach," *Frontiers in Energy Research*, vol. 10, 2023.

[17] A. Kalmikov, "Chapter 2 - wind power fundamentals," in *Wind Energy Engineering: A Handbook for Onshore and Offshore Wind Turbines*, ch. 2, pp. 17–24, Elsevier, 2017.

[18] M. Marquis, J. M. Wilczak, U. B. Gunturu, L. S. Sheridan, and K. L. McCaffrey, "Characterizing the spatiotemporal variability of wind power generation in the united states," *Journal of Applied Meteorology and Climatology*, vol. 59, no. 12, pp. 1899–1913, 2020.

[19] R. Nouri, A. Vasel-Be-Hagh, and C. L. Archer, "The coriolis force and the direction of rotation of the blades significantly affect the wake of wind turbines," *Applied Energy*, vol. 277, p. 115511, 2020.

[20] M. O. L. Hansen, *Aerodynamics of Wind Turbines*. London: Earthscan, 3rd ed., 2015.

[21] A. Lahouar and J. B. H. Slama, "Hour-ahead wind power forecast based on random forests," *Renewable Energy*, vol. 109, pp. 529–541, 2017.

[22] T. Göçmen, P. van der Laan, P.-E. Réthoré, A. P. Diaz, G. C. Larsen, and S. Ott, "Wind turbine wake models developed at the technical university of denmark: A review," *Renewable & Sustainable Energy Reviews*, vol. 60, pp. 752–769, 2016.

[23] M. Bastankhah and F. Porté-Agel, "A new analytical model for wind-turbine wakes," *Renewable Energy*, vol. 70, pp. 116–123, 2014.

[24] M. Mohandes, S. Rehman, N. Hilal, and F. H. Schulze, "Predictability of wind speed with heights using recurrent neural networks," in *2021 4th International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, pp. 1–5, IEEE, 2021.

[25] P. Andlinger, "Managing a grid when variable wind is prominent," *Andlinger Distillate*, 2019. Accessed: 2023-03-21.

[26] C. Hamon and M. Persson, "Wind power participation in frequency regulation: a profitability assessment for sweden," Tech. Rep. RISE Report 2022:42, RISE Research Institutes of Sweden, 2022.

[27] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[28] H. Z. Shiliang Sun, Zehui Cao and J. Zhao, "A survey of optimization methods from a machine learning perspective," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 10, pp. 1808–1835, 2021.

[29] F. Imam, P. Musilek, and M. Z. Reformat, "Parametric and nonparametric machine learning techniques for increasing power system reliability: A review," *Information*, vol. 15, no. 1, p. 37, 2024. Submission received: 12 November 2023 / Revised: 5 January 2024 / Accepted: 8 January 2024 / Published: 11 January 2024.

[30] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.
[31] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.
[32] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, ACM, 2016.
[33] G. E. Hinton, D. E. Rumelhart, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
[34] W. L. Hamilton, *Graph Representation Learning*, vol. 14. Synthesis Lectures on Artificial Intelligence and Machine Learning, 2020.
[35] G. Li, C. Xiong, A. K. Thabet, and B. Ghanem, "Deepergcn: All you need to train deeper gcns," *CoRR*, vol. abs/2006.07739, 2020.
[36] Open-Meteo, "Ecmwf weather forecast api documentation," 2025. ECMWF-based hourly forecast data for temperature and wind via Open-Meteo.
[37] National Renewable Energy Laboratory, "Flasc: A python library for floating-structure data analysis and visualization." `https://nrel.github.io/flasc/index.html`, 2025. Accessed: 2025-01-23.
[38] International Electrotechnical Commission, "IEC 61400-1:2005(E) Wind turbines – Part 1: Design requirements," 2005. Standard No. IEC 61400-1:2005(E).
[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, and É. Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[40] XGBoost Developers, "Multi-output regression with xgboost." `https://xgboost.readthedocs.io/en/stable/tutorials/multioutput.html`, n.d. Accessed: 2025-05-21.
[41] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.
[42] Z. Xu and Y. Li, "Learning physical simulation with message passing transformer," 2024. `https://doi.org/10.48550/arXiv.2406.06060`.
[43] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, pp. 41–75, 1997.

A

Appendix 1

A.1 Farm Layout and Weather Grid Points

The weather data from the IFS is provided on a grid. The blue points in Figure A.1 indicate the grid locations used to extract IFS weather variables, including wind speed and wind direction at 100 m height, as well as temperature and relative humidity at a 2m height. It is displayed relative to the wind farm and the wind turbines are marked in red. The IFS grid has a resolution of approximately 12 km, and data from 4 grid points were used.

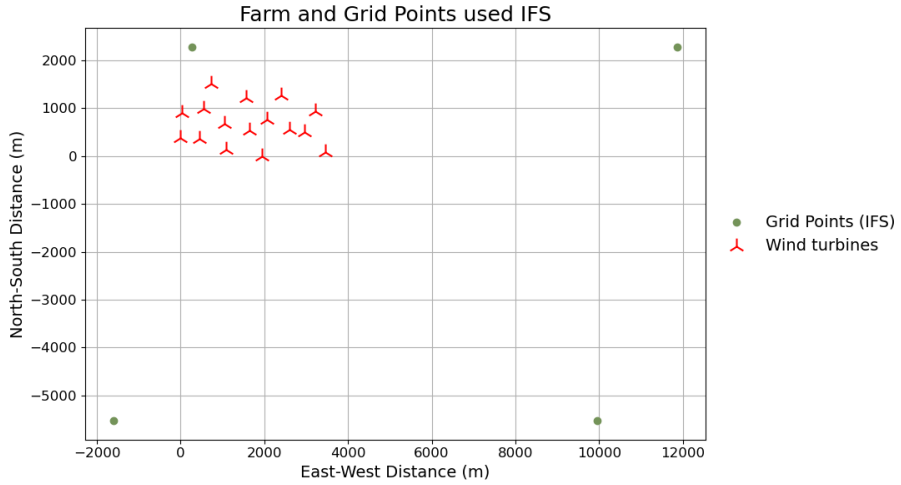

Figure A.1: Spatial distribution of IFS dataset grid points relative to wind turbine locations. Wind turbine positions are marked in red and grid points from IFS data set in blue [2].

Similarly, the weather data from MET Norway is also provided on a grid. Figure A.2 shows the selected grid points in blue, with turbines again marked in red. The MET Norway forecast has a finer spatial resolution of approximately 2.5 km, and data from 9 grid points were used.

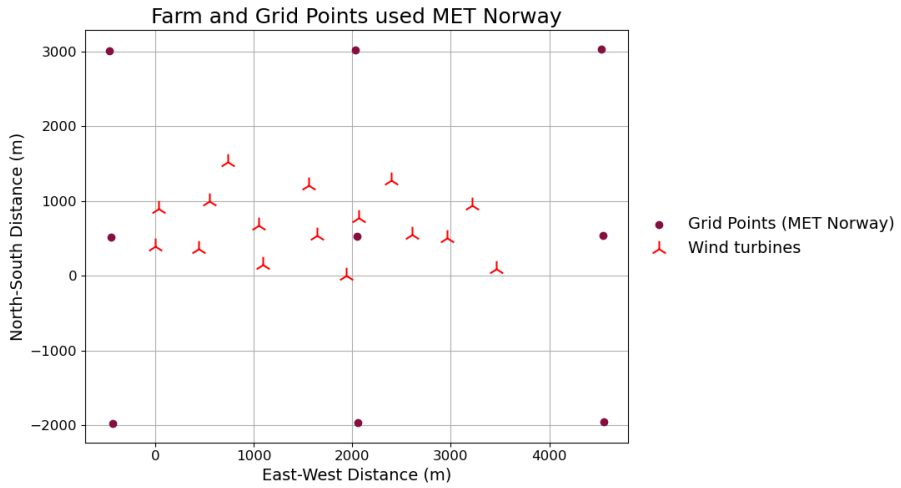

Figure A.2: Spatial distribution of MET Norway forecast data grid points used relative to wind turbine locations. Wind turbine positions are marked in red and grid points from MET Norway dataset in burgundy [3].

A.2 Data Cleaning

Before analysing the turbine data, a series of data cleaning steps were performed to ensure high data quality in the SCADA dataset. Table A.1 summarizes the percentage of data points removed for each turbine at each step of the cleaning process.

Data Cleaning Step	T00	T01	T02	T03	T04	T05	T06	T07
>7 turbines off	9.3	9.3	9.3	9.3	9.3	9.3	9.3	9.3
Curtailment	2.3	1.8	2.6	6.1	1.8	1.8	2.9	2.1
Outliers	0.6	0.6	0.5	0.6	0.6	0.6	0.6	0.7
Total	12.2	11.7	12.4	16.0	11.7	11.7	12.8	12.1

Data Cleaning Step	T08	T09	T10	T11	T12	T13	T14	T15
>7 turbines off	9.3	9.3	9.3	9.3	9.3	9.3	9.3	9.3
Curtailment	2.4	2.5	3.4	2.4	1.8	2.7	4.4	2.7
Outliers	0.8	0.8	0.5	0.7	0.7	0.7	0.5	1.0
Total	12.5	12.6	13.2	12.4	11.8	12.7	14.2	13.0

Table A.1: Percentage of data points removed per turbine and cleaning step. Cleaning steps are applied sequentially, meaning each step operates on the remaining data from the previous step. The total row represents the cumulative percentage of invalid data points per turbine.

Due to the removal of all data points containing invalid turbine values, the final cleaned dataset corresponds to 59.8 % of the original data. In the filtered dataset, it varies how frequently a turbine is operational, which is shown in Figure A.3.

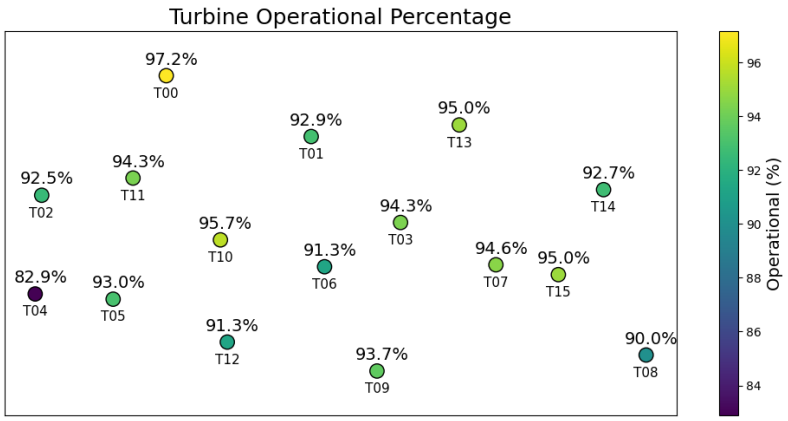

Figure A.3: Percentage of time turbine is operational after filtering.

A.3 Data Analysis

A.3.1 North Calibration

Figure A.4 illustrates wind direction measurements across all turbines before and after north calibration. Prior to calibration, large discrepancies between turbines are observed due to inconsistent sensor orientations. After calibration, the wind directions are aligned more closely, indicating improved consistency and a more accurate representation of the global wind direction. It is noticeable that certain turbines, such as Turbine 0 and Turbine 2, consistently measure a lower wind direction than the others, even after north calibration. This suggests residual offsets may remain in the sensor alignment or calibration process for these specific turbines.

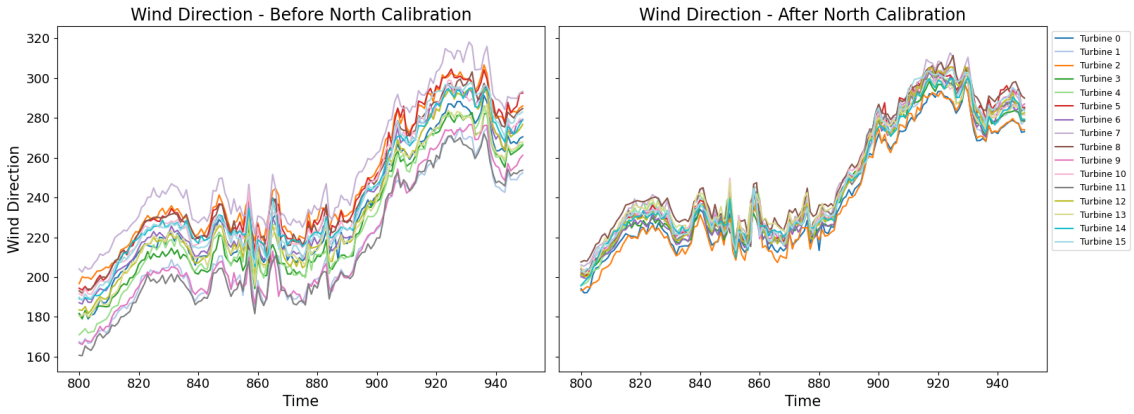

Figure A.4: Wind direction measurements for all turbines over a randomly selected 150-point time segment, starting at index 800, before and after north calibration. The left plot shows uncalibrated wind direction signals, where significant offsets between turbines are visible. The right plot shows the same measurements after north calibration, with improved alignment between turbines.

A.3.2 Comparing IFS and UGW wind speed data

To assess how the wind speed data from IFS and UGW compares to each other Figure A.5 shows the wind speed from IFS and UGW at each time. From the figure it is clear that the wind speed from UGW is generally higher than the wind speed in the IFS dataset.

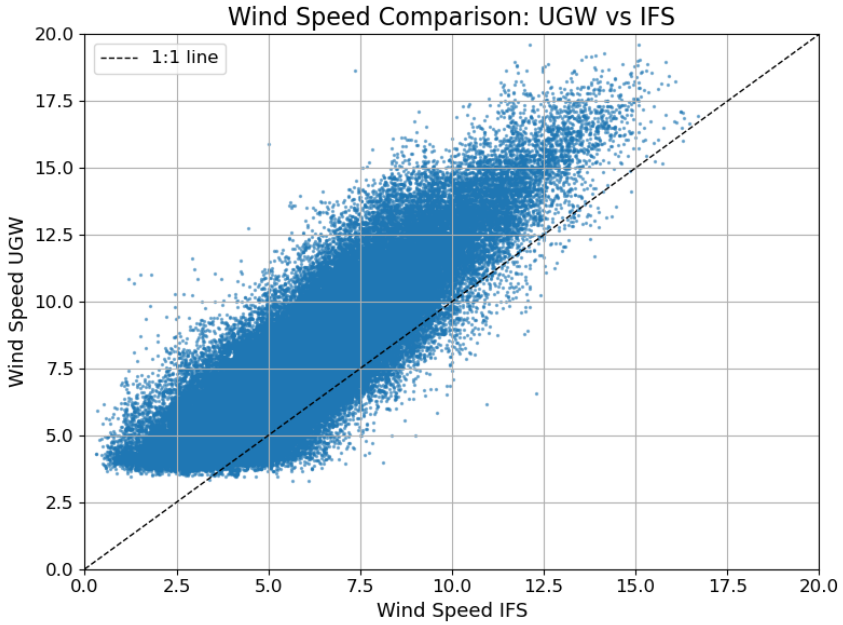

Figure A.5: Comparison of wind speed data from IFS and UGW dataset. The identity line (1:1 line) is shown in black. The windspeed from UGW is generally higher than the wind speed in IFS.

A.3.3 Wind speed variation

The models uses a global wind speed and wind direction as input for the whole farm. However, spatial variability in wind speed is critical to accurately capture variations in power output across the farm. To get an understanding of the variability of the wind speed over the farm Figure A.6 displays the maximum difference in wind speed between upstream turbines. This is done by first filtering the data for the dominant wind direction, which is 250°. Then the difference is calculated for the 2 turbines with the maximum wind speed, which were used to create the UGW dataset. The resulting distribution is shown in orange. The upstream turbines were also manually selected based on the wind direction of 250° (turbines 02, 04, 08, 09, 12) and the maximum difference between the selected turbines was calculated. The resulting distribution is shown in dark blue.

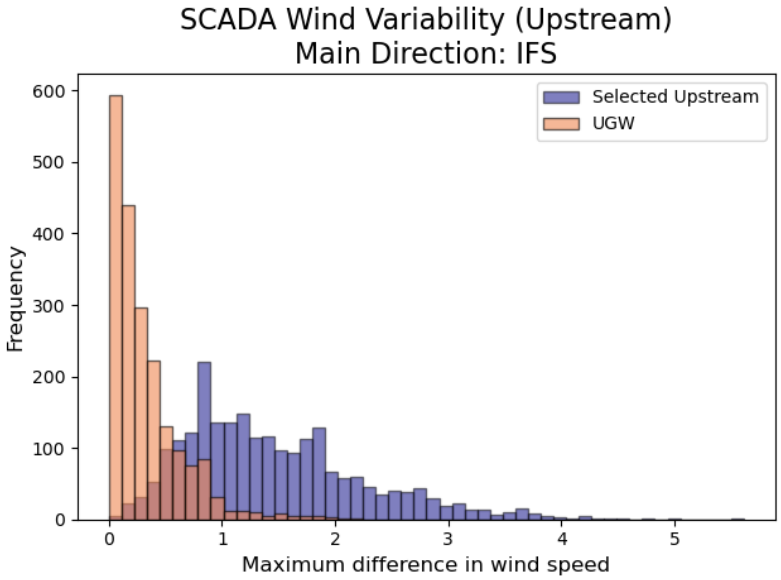

Figure A.6: Variability of wind speed from upstream turbines in main wind direction 250°. Main wind direction was given using IFS dataset.

The figure shows that the variability in the selected upstream turbines is significantly higher than the two maximum turbines. Indicating that the wind speed varies across the farm and a global wind speed might not provide accurate enough information to the models.

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY