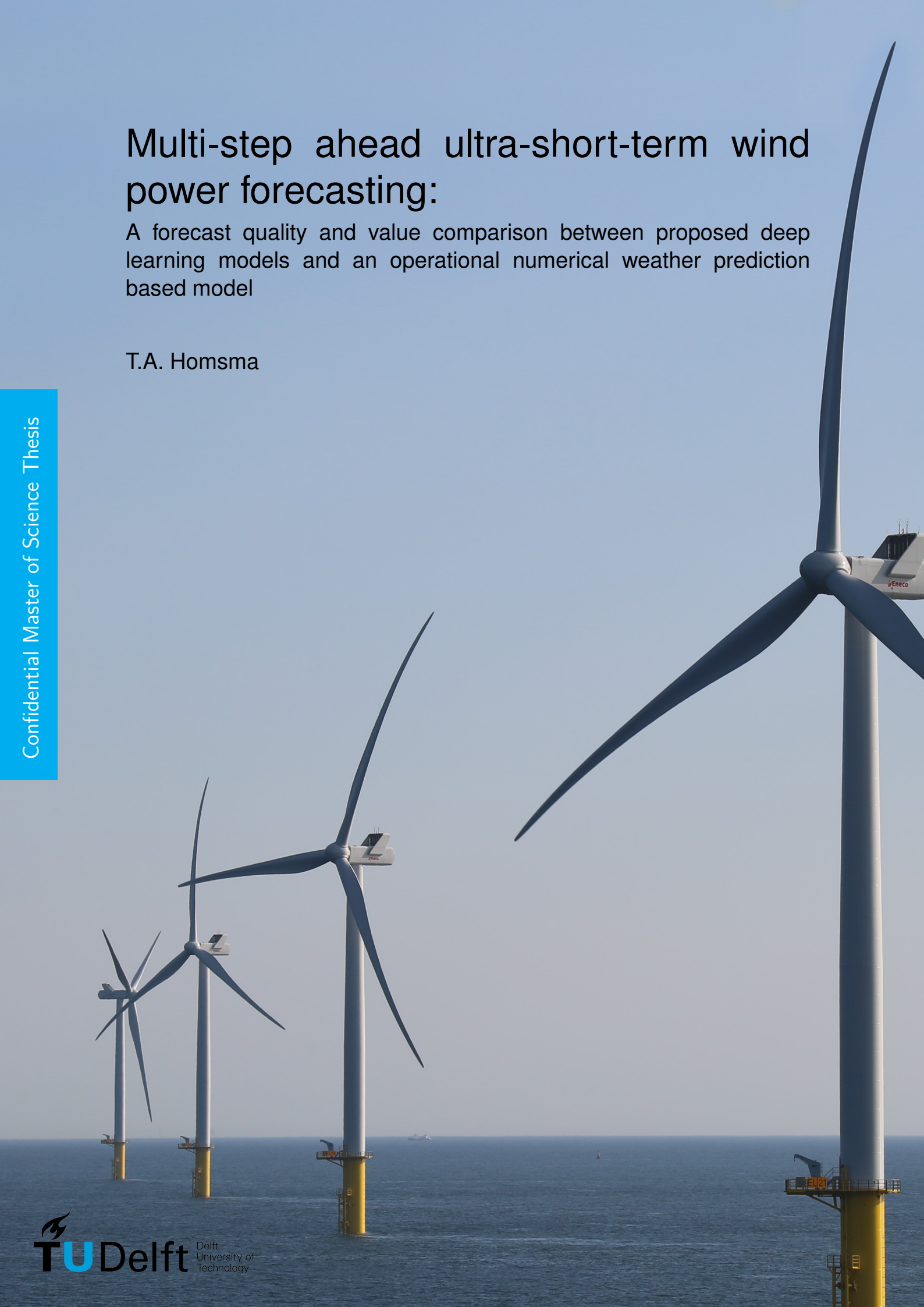
Multi-step ahead ultra-short-term wind power forecasting:

A forecast quality and value comparison between proposed deep learning models and an operational numerical weather prediction based model

T.A. Homsma

Confidential Master of Science Thesis



TU Delft Delft University of Technology

Multi-step ahead ultra-short-term wind power forecasting:

A forecast quality and value comparison between proposed deep learning models and an operational numerical weather prediction based model

Master of Science Thesis

For the degree of Master of Science in Sustainable Energy Technology at Delft University of Technology

by

T.A. Homsma

To be defended publicly on Wednesday, August 18, 2021 at 14:00.

Student number:	4375084	
Project duration:	December 1, 2020 - August 18, 2021	
Thesis committee:	Prof. dr. S.J. Watson,	TU Delft (AE) Supervisor
	Dr. S. Basu,	TU Delft (CiTG) Supervisor
	MSc. V. Visser,	Eneco Supervisor

Cover image courtesy: Eneco

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Eneco, one of the leading utility companies in The Netherlands, supported the work in this thesis. Their cooperation is gratefully acknowledged.

Abstract

The ongoing large scale adoption of wind power increases the associated risks related to the variability. An essential way to mitigate these risks is to forecast production accurately. Because of its commercial and technical relevance ultra-short-term wind power forecasting (UST-WPF) is the chosen forecast horizon of this thesis. Two open areas of research influenced the direction of this research. Firstly, the desire for more insight into the practical application of forecast methods considering both forecast accuracy and value. Secondly, the application of deep learning methodologies in the field of forecasting. Therefore, the research goal of this project has been to create insight into the potential of deep learning models for both forecast quality and value on the UST-WPF horizon.

The status quo at Eneco for UST-WPF is a numerical weather prediction (NWP) based model with a rudimentary ultra-short-term (UST) correction with real-time power data. The methodology followed was the development of four UST-WPF models for Princess Amalia Wind Farm (PAWP) with a 16 programme time unit (PTU) forecast horizon and a forecast frequency of 1 PTU. Both model 1 and model 2 only use real-time data and are based on a multilayer perceptron (MLP) and a long-short-term memory (LSTM) architecture, respectively. After optimisation, these models were trained ten times to compute the 10th percentile, median and 90th percentile forecast. The other two proposed models are a multivariate combination of the median ensemble forecast models with the Eneco model.

The accuracy of the models was compared to two benchmark models: a Persistence and the Eneco model. Additionally, a novel framework was designed to evaluate the forecast value relative to the Eneco model on a variable forecast horizon.

The forecast quality results show that the models based on real-time data outperform the Persistence model on a nine PTU ahead horizon, and the multivariate combination models outperform the Eneco model on a nine PTU ahead horizon. The difference in performance between the MLP and LSTM is minimal in the proposed configurations. However, the LSTM model does show to be more robust compared to the MLP model. The forecast value results show that all proposed models generate positive value relative to the Eneco model, but the statistically best model does not necessarily generate the most value. To summarise, the results indicate that the proposed deep learning models can contribute both in quality and value up to 9 PTUs ahead.

Even though these results are encouraging, there are still multiple considerations; for example, the model still needs to be evaluated for a whole year. Future research recommendations are but not limited to: explore novel architectures (e.g., encoder-decoder model), include NWP data in features, and research the explainability of deep learning time series models.

Table of Contents

List of Figures

List of Tables

Acknowledgements

First of all, I would like to address that I am very grateful for the opportunity to graduate from Delft University of Technology (TU Delft) at Eneco Energy Trade. When I reached out to both Paul Smeets and Willem Willems in mid-November 2020, they were able to help me formulate a research topic and arrange the contractual formalities on very short notice. Because of this, even amid the Covid-19 pandemic, I could start this project in December 2020.

I want to thank my TU Delft supervisors Simon J. Watson and Sukanta Basu, for taking the time and making an effort to guide me through this process. Simon's continuous support and extensive experience in wind energy research have lifted the quality of this thesis. Sukanta's positive energy, drive, knowledge of deep learning methods and openness make him an ideal mentor and daily supervisor. This work would not have been possible without their encouragement and advice.

In addition, I would like to thank all colleagues at Eneco who have been involved in this project. Especially my supervisors Truusje Quak and Vincent Visser. Truusje has introduced me to Eneco's relevant data, the current models, and operational forecasting processes. Vincent has been of tremendous help in defining forecast value and offering his advice on the project based on his experience as a data scientist. Furthermore, I would like to thank Mathias Veenman for answering any programming question and Aatish Kumar for teaching me how to work on the Data Science Virtual Machine (DSVM).

Finally, I am forever grateful to my family, especially my parents and sister: Karin Homsma-Dekker, Tjeerd Homsma and Eva Homsma, for their emotional, financial, and rational support for the last 25 years. Also, I would like to thank Maud van den Berg, my girlfriend, for the exceptional care, love and faith. Their unconditional love and support have given me the energy to follow my interests and finish this thesis.

Delft, University of Technology
July 27, 2021

T.A. Homsma

"Those who have knowledge don't predict.
Those who predict don't have knowledge."

— *Lao Tzu*

Chapter 1

Introduction

The introductory chapter briefly discusses the context, relevance and scope of this thesis. Furthermore, the structure of the report is provided to guide the reader through this work.

1-1 Context

The growing interest to mitigate climate change and reduce carbon emissions has stimulated the implementation of variable renewable energy (VRE) sources, in particular solar photovoltaics (PV) and wind energy technologies [40]. Over the last decade, there has been a steady increase in Europe's installed wind power capacity, and even the recent global Covid-19 crisis has not broken this trend. In 2020 Europe's installed wind power capacity increased by 14.7 GW, and an increasing installation rate is expected in the year to come, see Figure 1-1 [22, 44].

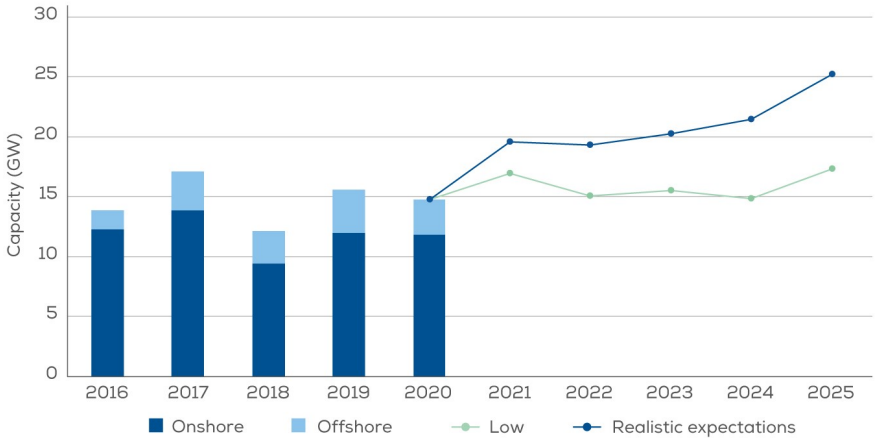


Figure 1-1: Past and projected annual wind power capacity of EU27+UK [44].

The rapid increase in installed wind power capacity is associated with economic and environmental benefits, but it also comes with multiple challenges. The major challenges are related to the intermittent nature of wind power, caused by meteorological fluctuations [1, 28]. The variability in supply can lead to a large imbalance, which in turn can significantly impact market prices and grid stability [21]. Taking this risk into account can form an obstacle for further wind power penetration within the power system. However, large-scale studies in multiple countries have shown that accurate forecasting systems can mitigate these obstacles [15, 21].

The underlying importance of forecasting from the perspective of a utility is explained as follows. TenneT, the Dutch transmission system operator (TSO), is the responsible party for electricity transport and grid stability on a national level. To fulfil this role, Balancing Responsible Parties (BRPs) like Eneco are obliged to update TenneT on their scheduled production, consumption and transportation needs. BRPs are financially responsible for the difference between their actual production and consumption compared to the reported schedule unless they have already corrected their position in the market. The reported schedule and adjustments in the market are based on forecasting models. In this regard, the forecast quality influences the potential revenue of BRPs [47].

Variations of wind happen on all time scales, but only from minutes to weeks is considered relevant for the BRPs' wind power forecast [13]. The shortest forecast time scale, ultra-short-term wind power forecasting (UST-WPF), needs the highest accuracy and is considered the most difficult because of the 15-minute temporal resolution and 4-hour forecast horizon [28]. Currently, Eneco operates a numerical weather prediction (NWP) based forecast model on their wind farms with a rudimentary ultra-short-term (UST) correction for the 4-hour forecast horizon. Given the fact that UST-WPF finds its application in many areas including but not limited to short-term trading, asset curtailment and commitment of quick-start resources make it a commercially important forecast horizon [7, 15, 21, 28, 39]. To summarise, both from a technical and commercial point of view UST-WPF is an interesting area of research, which will become even more important with the expected increase in installed capacity. Therefore, UST-WPF will be the focus of this thesis.

1-2 Problem statement

The importance of forecast techniques to mitigate the challenges related to the intermittent nature of wind power motivated the conduction of a literature study. Based on the literature review into UST-WPF, the following two research topics were selected.

1-2-1 Deep learning methods

The increasing amount of computational resources and data have made deep learning methods more common. However, deep learning is still a relatively new area of research with many recent publications. Contributions to this field can be both through introducing novel methods as well as through validation [7]. In this respect, the following text is worth quoting.

> "Deep learning for time series is a relatively new endeavour, but it's a promising one. Because deep learning is a highly flexible technique, it can be advantageous

> for time series analysis. Most promisingly, it offers the possibility of modelling highly complex and nonlinear temporal behaviour without having to guess at functional forms. Deep learning has not yet delivered the amazing results for forecasting that it has for other areas, such as image processing and natural language processing. However, there is good reason to be optimistic that deep learning will eventually improve the art of forecasting while also lessening the brittle and highly uniform nature of assumptions and technical requirements common for traditional forecasting models."
>
> — *Aileen Nielsen [37]*

1-2-2 Forecast value

In the studied literature, the main focus is often on theory and lowering a single statistical error metric. However, the single error metric indicates the forecast quality and not necessarily of the forecast value [2]. Therefore, it is interesting to shift the focus towards practical application of forecast methods considering both accuracy and value [7, 13].

> "R&D will remain important in the future for improving not only the accuracy of the forecasts but also their value."
>
> — *Gregor Giebel [13]*

1-3 Objectives and research questions

The main objectives of the thesis are to:

- Create insight in the potential of deep learning models for UST-WPF.
- Create insight in forecast value from the perspective of a utility.

In order to achieve the objectives mentioned above, the following research questions have been formulated:

1. What are the current state-of-the-art forecast techniques for UST-WPF?
2. What set of input variables are the most relevant on the UST time scale?
3. What are the currently applied loss functions and error metrics?
4. How does UST-WPF based on deep learning models compare to the currently applied UST correction NWP based model and a naive forecasting model?
5. How does the forecast value defined by Eneco compare with the performance on the chosen standard error metric?

1-4 Outline

The report is structured in the following manner:

- Chapter 1: (this chapter) explains the relevance of the research and states the objectives and research questions.
- Chapter 2: is a literature review that provides background on wind power forecasting and time series forecasting domain knowledge, both in the context of the project scope defined in Chapter 1. Subsequently, the state of the art on UST-WPF is summarised and discussed.
- Chapter 3: offers insight into the methodological decisions made during the data collection, model development and evaluation of the results. Furthermore, it presents information on the applied devices and software.
- Chapter 4: displays and explains how to interpret the results from the described test setup in Chapter 3. Moreover, the meaning, implications and limitations of the results are discussed.
- Chapter 5: reflects on the research and summarises the main conclusions concerning the research questions. Moreover, future recommendations are given based on the developed knowledge of this work.

Chapter 2

Literature review

The first subsection gives background on wind power forecasting (WPF) within the structure of the recently published International Electrotechnical Commission (IEC) classification framework [21]. Subsequently, the second subsection provides the reader with the necessary background on time series forecasting. Finally, the state of the art literature on ultra-short-term wind power forecasting (UST-WPF) is summarised.

2-1 Wind power forecasting

In the past two decades, the research interest in WPF has grown significantly [21, 39]. There used to be no universal standard to classify the research within this large domain. Therefore, the definition of labels within classifications varies between researchers, organisations and countries. However, the recently published IEC standard has put the lack of consensus about classification within the scientific community to rest.

The IEC classification consists of five categories: the time scale, spatial range, input data, forecasting model and forecasting form. Any forecast solution can be labelled for every classification; nevertheless, some labels can be considered mutually exclusive (e.g., medium-term forecasts require numerical weather prediction (NWP) data as input) [21].

Table 2-1 summarises the classification of this thesis within the IEC framework. Because the time scale and spatial range were predetermined based on research and commercial interest, only the other three classifications of WPF models are discussed in more detail in the following subsections.

Table 2-1: Characteristics of this research within the IEC framework [21].

Classifcation standard	Type
Time scale	0.25-4 hours (ultra-short-term)
Spatial range	Wind farm
Modelling input data	Input data without NWP data
Forecasting method	Persistence method Statistical and Learning methods Multivariate combination methods
Forecasting form	Deterministic forecasting

2-1-1 Classification based on modelling input data

Input data without NWP data

A model in this category uses supervisory control and data acquisition (SCADA) data; this includes real-time and historical data from the wind turbine(s). Additionally, data from other wind turbines or measurement points can be included [1]. The most common meteorological input features are wind speed, wind direction and temperature. Less frequent features are atmospheric pressure, and relative humidity [15, 33]. Zhou et al. [47] reported the correlation of these meteorological features with wind power for his data set, see Table 2-2 [47]. In the context of UST-WPF some discrepancy or ambiguity can be found in the literature. Some sources state that NWP based models start to outperform time series models from 3-6 hours lead-time onward, while others state that NWP data adds value from 1-2 hours onward [17, 21, 45].

Table 2-2: Correlation coefficient between meteorological features and wind power [47].

Meteorological factor	Correlation coefficient
Wind speed	0.62
Wind direction	0.29
Pressure	0.21
Temperature	0.07
Humidity	0.01

Input data with NWP data

The forecasted wind behaviour and other atmospheric properties from NWP models can function as inputs for WPF models. Together with other information (e.g., topographic information) the Reynolds-averaged Navier–Stokes (RANS) equations form the basis of NWP models [17, 21]. The RANS equations are partial differential equations for which no analytic solution exists; thus, one must rely on numerical solvers. Running these models is computationally expensive; therefore, update frequencies are generally between 1-12 hours, depending

on the model. The results of these models are essential for WPF models with longer forecast horizons.

2-1-2 Classification based on forecasting methods

Forecasting methods can be divided into persistence methods, physical methods, statistical and learning methods, and multivariate combination methods [15, 21].

Persistence methods

The persistence method is generally used as a benchmark method. It takes the current power measurement as the forecasted value for the next time step(s) [10]. This method is only suitable for short time scales, and the accuracy can deteriorate quickly for longer forecast horizons [21]. The main advantage is that this is a very simplistic model that requires no external variables [15].

Physical models

These models can predict the power production based on physical information about the wind farm and NWP data. Physical models have proven to be very successful for forecasting with a time horizon of more than 4 hours and find their application predominantly in short-term (i.e., day ahead) and medium-term (i.e., days to a week ahead) renewable power forecasting (RPF) [21]. The main advantage is that physical models do not require lengthy historical data. On the other hand, the accuracy of these models strongly depends on the topography of the location [10]. Furthermore, these models are computationally expensive and relatively complex [15].

Statistical and learning methods

As opposed to physical models, statistical models do not include physical processes. Statistical models apply a direct transformation from the input variables to wind power. The input variables can consist of both historical power production and NWP data [1]. In the case of UST-WPF, real-time data is essential [21]. These methods are easy to model and generally perform very well for UST-WPF [15]. Statistical methods can be subdivided into time series based and learning methods.

Time-series based A time series is a chronological set of observations of a variable. In the case of a regular pattern, past values can be used to predict future values through a function [1]. The most common are the methods proposed by George Box and Gwilym Jenkins, referred to as the Box-Jenkins methods [4, 15]. In the studied literature, multiple variations and extensions of the autoregressive integrated moving average (ARIMA) model have been tried to optimally fit a model to a time series [10]. Discussing all these alternative forms exceeds the purpose of this section; therefore, only the primary model components are explained.

The ARIMA consists of three components:

- **AutoRegressive (AR)**, is the autoregression of a specified number of lagged values.
- **Integrated (I)**, is the differencing (i.e., subtracting consecutive values) of the time series to transform the raw input into a time series that consists of the deltas between consecutive values.
- **Moving Average (MA)**, is the autorregression of the lagged residual errors. Correcting the forecast with a predicted residual error can improve the model.

The number of lagged values for AR and MA are given as hyperparameters p and q, respectively. The order of differencing (i.e., how many times the differencing procedure is executed) is hyperparameter d.

The advantage of this model is that it is fairly easy to implement. Furthermore, it requires minimal computational capacity [7]. However, without modification, it cannot include exogenous variables. Moreover, learning methods tend to outperform this model on large non-stationary time series [6, 15, 35].

Learning methods Artificial intelligence (AI) methods find their application on all forecast horizons [21]. The four categories of AI methods are linear machine learning, nonlinear machine learning, ensemble machine learning and deep learning. The aggregate of different models within these categories is more than 50; therefore, the overview in Table 2-3 should be considered non-exhaustive. Additionally, hybrid models exist that either combine multiple learning methods (e.g., CNN-LSTM), combine decomposition techniques and learning methods (e.g., WT-ANN and EMD-SVM), physical and learning methods, or statistical and learning methods [7, 10].

Table 2-3: A general overview of state-of-the-art learning methods.

Linear Machine learning	Nonlinear machine learning	Ensemble machine learning	Deep learning
Linear regression	KNN	Random forest	MLP
	Decision trees	Gradient boosting	CNN
	Support vector regression	Stacking	LSTM

Multivariate combination methods

The multivariate combination methods are a weighted average of the earlier discussed models [10, 21]. The objective of this method is to reduce the forecast error by incorporating the positive characteristics of different models [15]. Another advantage is that multivariate combination methods are usually more robust.

2-1-3 Classification based on forecasting form

Deterministic forecasting

This form provides a single power value for every forecasted time step. The accuracy is generally high, but the uncertainty related to the forecast is not provided [21]. Therefore, the evaluation of deterministic forecasts is more straightforward.

Probabilistic forecasting

Probabilistic forecasting quantitatively provides the probability related to the forecasted values. It aims to represent the uncertainty related to atmospheric conditions. The four most common methods are statistical methods, statistical scenarios, physically-based ensemble forecasts and perturbation-based forecasts [16, 21]. The first two methods create a probabilistic forecast from a deterministic simulation. The latter two apply either multiple NWP models or apply varying input conditions, respectively. Consequently, these methods are computationally more expensive than statistical methods. There is increasing research interest in this field as it can improve situational awareness and consequently improve the decision making [16, 34, 36]. The forecast can be in the form of quantiles, ensembles and parametric distributions [34].

Event forecasting

This form aims to predict the probability of an event's occurrence (e.g., significant ramp event or cut-out situations) [21]. The forecast user is notified if the probability exceeds the set threshold [34].

2-2 Time series forecasting

This section provides the reader with the necessary background on time series forecasting and the models applied for this research. The structure corresponds roughly to Chapter 3.

2-2-1 Input data types and wrangling

For time series forecasting, the input data quality is of utmost importance. However, the gathering of high-quality data is often a bumpy road with many potential pitfalls. The first section explains the various types of input data. Subsequently, the second section discusses the data wrangling process.

Input data types

Section 2-1-1 explains the classification of WPF based on input data. Here a more generic taxonomy is provided that can describe any type of input when forecasting time series [32, 37].

- **Univariate data**, where one variable is measured over time (e.g., wind power at one wind farm).
- **Multivariate data**, where multiple different variables are measured over time within a single experimental unit (e.g., wind power and wind speed at one wind farm).
- **Panel data**, where the same kind of univariate or multivariate data is measured over time at multiple independent instances (e.g., wind power at multiple wind farms).
- **Metadata**, information about other data (e.g., day of the week corresponding to the wind power measurements).

Input data wrangling

Data wrangling, the process of preparing data for downstream purposes, can be a time-consuming operation but crucial for the model's performance. The following section will discuss time series wrangling, specifically concerning timestamps and data cleaning.

Timestamps provide helpful information to the data and make time-series analysis more intuitive. Nevertheless, interpreting time series should be done with great precaution. Since in practice, it is not always clear what the timestamp represents. To illustrate the issue, when encountering data with a fifteen-minute frequency, does it represent the measurement at that specific point in time, or is it the mean of data gathered at every minute?

Another issue quintessential to timestamps is related to time zones. Even though the database convention is to store data in Coordinated Universal Time (UTC) instead of local time, when combining data from multiple sources, this should not be assumed as it can result in past data at future timestamps or future data at past timestamps (i.e., lookahead). The fundamental trouble is that there is no generic methodology or test that can detect lookahead [37]. Therefore, the only way to prevent mistakes related to timestamps is to continuously be aware and critical towards obtained results and consult documentation when available.

After establishing what the data represents, the next step is to clean the data. This section does not cover all the possible data cleaning techniques but aims to describe a general framework based on the studied literature [28, 37].

1. **Missing data**, reasons for missing data can either be systematic (e.g., when the system availability of a wind farm is zero, no wind speed measurements are gathered) or random (e.g., an unexpected software update has hindered the data writing process). Information about the nature of the missing data can be valuable. There are two types of missing data in time series:

   (a) **No observation**, the timestamp exists, but there is no observation. The absent observation is, in most cases, filled with a Not a Number (NaN) value.

   (b) **Missing timestamp**, the timestamp does not exist, which automatically results in no observation. Missing timestamps commonly originate from a Structured Query Language (SQL) database, which does not have time as a privileged information axis. As SQL finds its origin in transaction data storage, where time is just one of the many primary keys. A solution to obtain a regular time series is to insert these missing datetimes with NaN observations.

2. **Anomaly detection**, identify which data points are out of the ordinary based on analysis and domain knowledge (e.g., a wind farm producing more than its maximal theoretical production capacity). The most straightforward approach is to interpret these instances as missing. However, except when there is an explanation for the anomalies, a more intelligent approach can be desirable.

3. **Solutions to missing data**, there are multiple techniques to fill or circumvent the missing data. Nevertheless, it can be illogical to proceed with inadequate data, considering that even the most sophisticated techniques available to ameliorate the data have their limitations.

(a) **Downsample**, if the downstream use case allows it, downsampling can be a straightforward method to reduce the amount of missing data (e.g., take the mean of all per minute data within one hour).

(b) **Imputation**

i. **Data from another source**, when the data from another source is statistically similar enough (e.g., wind speeds measured at a nearby wind farm), then imputation from this data set can be an option.

ii. **Forward fill**, where the most recent value is carried forward to fill the missing observations. The advantages of the forward fill method are low computational demand, applicable to online data, and the imputation quality for a limited amount of consecutive missing values is relatively high.

iii. **Moving average**, comparable to forward fill, but uses multiple recent values to compute an average, which is carried forward to fill the missing observations.

iv. **Interpolation**, there are many interpolation techniques, for example, linear, polynomial and spline. Generally, these perform better than the forward fill and moving average methods, but interpolation often includes lookahead, making it undesirable for forecasting projects.

(c) **Remove timestamps with no observation**, this is the most time effective, but not all models perform well with irregular input.

2-2-2 Time series characteristics and exploratory data analysis

Developing insight into the properties of the available data can be a valuable undertaking before preprocessing and modelling. In this section the most common time-series properties and exploration techniques are described [5, 37].

The most important properties to explore are:

- **Correlation:** indicates the degree to which two variables move in relation to each other, where 1 indicates a strong positive correlation and -1 a strong negative correlation.
- **Autocorrelation:** is the correlation of a specific signal with a time-shifted copy of the same signal as a function of the lag.
- **Partial autocorrelation:** is the autocorrelation that has been corrected for the indirect correlation between the original and the time-shifted copy of the signal. Therefore, the partial autocorrelation reveals the direct correlation and indicates which lagged values truly contain information. This information can indicate the number of meaningful lag values.
- **Stationarity:** refers to the case when the statistical properties of the time series are not time-dependent. The most important characteristics are constant mean over time, constant variance over time and no seasonal component. In practice, it is generally easier to falsify the stationarity hypothesis than to find definite proof.
- **White noise:** a time series where all variables are independent and identically distributed (IID) with zero mean and constant variance, which implies zero correlation between the variables. Therefore, the series is considered unpredictable.

- **Random walk:** a time series where the next value is a random modification of the previous value and consequently is considered unpredictable.
- **Cyclical behaviour:** recurring behaviour without a fixed period.
- **Seasonality:** recurring behaviour with a constant frequency (e.g., diurnal, monthly, yearly frequency).
- **Trend:** is the increase or decrease of the mean over a more extended period.

In order to inspect these properties, a common approach is to visualise the data. Examples include line plots, histograms, Box and Whisker plots, heat maps, scatter plots, decomposition plots and autocorrelation plots. For illustrations of these techniques, see subsection 3-1-2 where most of these visualisation techniques are applied.

Another option is to check the time dependency of summary statistics. For stationarity specifically, it is often desirable to further explore the data through hypothesis tests. The most widely used hypothesis test is the Augmented Dickey-Fuller (ADF) test. The null hypothesis is that the time series has a unit root and is thus non-stationary. This hypothesis can be rejected for a specific significance level based on the results (i.e., the p-value of the test). Nevertheless, even when rejected, this test cannot be considered definitive proof for stationarity as this test has a relatively high type I error (i.e., false-positive results).

2-2-3 Pre-processing data into features

The insight gained from data exploration can be applied to improve the input data. This section first discusses three methods to modify the data: transforms, moving average smoothing, and decomposition. The final subsection explains the various feature engineering options.

Transforms

Many models tend to converge more quickly and consistently after the data has been scaled or standardised. The first changes the data range but keeps the distribution intact (e.g., Min-Max scaler see Eq. (2-1)). While the latter often implicitly scales the data when the distribution is changed to have a standard deviation of one (e.g., standard scaler see Eq. (2-2)). Another difference between these methods is that the standard scaler does keep the signs of the data, while the Min-Max scaler transforms the data to only positive values [23].

$$x = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{2-1}$$

$$x = \frac{x - \mu}{\sigma} \tag{2-2}$$

For signals with a strong trend power transforms can scale the data. The square root transform linearises time series with quadratic growth. Time series with an exponential trend and only positive data require log transformation. In practice, these perfect theoretical trends are rare. A more variable solution is the Box-Cox transform capable of finding a more optimal

transformation for the time series by tuning lambda. Eq. (2-3) shows the general form of the Box-Cox transform. The hyperparameter lambda ranges between -5 and 5, determining the type of transformation.

$$y_i^{(\lambda)} = \begin{cases} \frac{y_i^{\lambda}-1}{\lambda} & \text{if } \lambda \neq 0 \\ \ln y_i & \text{if } \lambda = 0 \end{cases} \tag{2-3}$$

Moving Average Smoothing

Moving average smoothing aims to reduce noise by taking the average of a specific number of subsequent values and sliding this data window over the time series. The output is a new time series with reduced high-frequency components.

Decomposition

The decomposition of a time series is the separation of the signal into multiple components. These components can be systematic (i.e., deterministic) or non-systematic (i.e., non-deterministic). The difference is that the systematic components contain structure. Therefore, the systematic components can be forecasted separately and recombined to give the forecast of the original signal [10].

Classical time series decomposition assumes a signal to contain a level and noise component (i.e., the non-systematic component). Additionally, a signal can contain a trend and seasonality/cyclical component. This type of decomposition technique isolates the trend and then the seasonality component through computing moving averages. The disadvantage of this method is that two hyperparameters have to be predetermined. Firstly, the periodicity of the signal. Secondly, the relationship between the components, which is either additive or multiplicative.

Alternative decomposition methods are based on time-frequency analysis. The reason for both time and frequency analysis is to prevent the loss of information. The time-domain representation loses frequency resolution, and frequency-domain representation loses time resolution. To illustrate, in frequency domain analysis, the Fourier transform provides magnitude information as a function of frequency, but not when in time that frequency occurs, which is relevant information for non-stationary signals. This concept is visualised in Figure 2-1. There are two categories of time-frequency methods: adaptive or with an a priori basis function.

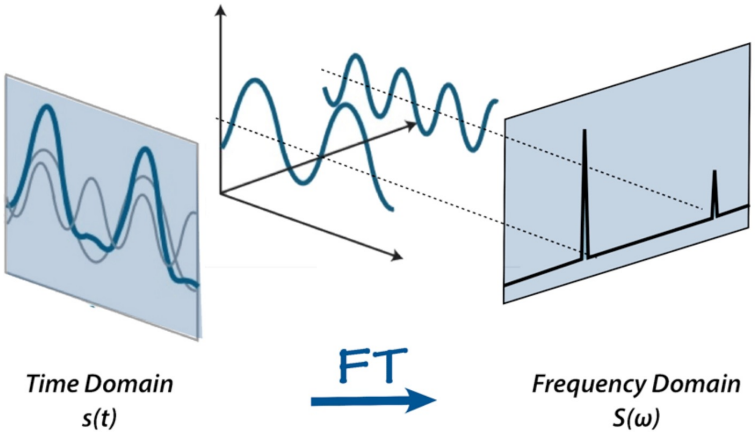


Figure 2-1: Illustration of a time series Fourier transform [12].

The most common time-frequency method with an a priori basis function is the wavelet transform (WT). WT has a very similar working principle compared to Fourier transform but has a different basis function. Fourier transforms and WT are computed with Eq. (2-4) and Eq. (2-5), respectively. The Fourier transform basis function consists of infinite sine and cosine waves, while the WT uses a compact support wavelet signal (ψ). The wavelet signal is adjusted by the scaling factor (a) and time shift factor (b). That scales the frequency and move the basis function over the signal, respectively. This method is cable of analysing non-stationary linear signals because of the time shift factor.

$$X(f) = \int_{-\infty}^{+\infty} x(t) e^{-i2\pi ft} dt \tag{2-4}$$

$$T(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \psi^*(\frac{t-b}{a}) dt \tag{2-5}$$

Examples of adaptive methods that can decompose non-stationary non-linear signals are empirical mode decomposition (EMD) and variational mode decomposition (VMD). EMD has an empirical and VMD has a theoretical base [11, 19]. For wind power forecasting EMD and VMD have shown better results in recent years compared to papers that have applied WT [14, 20, 47].

The EMD method decomposes the signal in intrinsic mode functions (IMFs) through the sifting process. IMFs can have a variable frequency and amplitude, but have to satisfy two characteristics. The first requirement is the same number of extrema (i.e., maxima and minima) and zero crossings. Secondly, the combination of the cubic spline through the maxima and the minima should be zero mean. Ensemble empirical mode decomposition (EEMD) is a modified version of EMD that reduces the mode mixing issue related to EMD through ensembles. To every ensemble, a different white noise signal is added. The output of EEMD is the ensemble mean of the corresponding IMFs.

The VMD algorithm was invented in 2014 to solve EMD's sensitivity to noise and sampling. The mathematical and theoretical background are outside the scope of this thesis, but the

performance of the decomposition method is strongly related to the number of modes hyper-parameter. An excessively high number of modes increases the computational complexity. Figure 2-2 illustrates the results of VMD on a wind power signal.

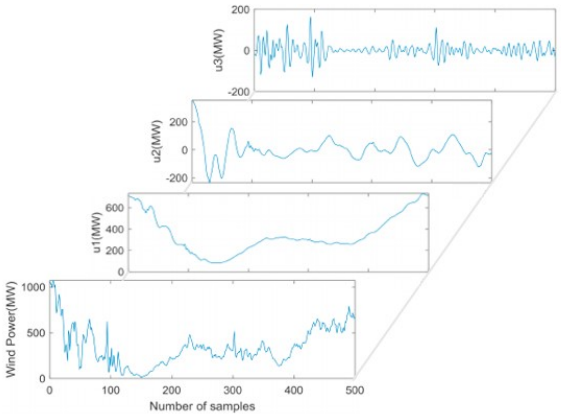


Figure 2-2: Illustration of VMD for a wind power signal [14].

Feature engineering

The final step of pre-processing is feature engineering. The features are the inputs of the model. For time series forecasting, there are four categories of features [5]:

- **Time-based features**, provide information about the time (e.g., hour of the day or season of the year). Approaches to design these features are:
  - Sine and cosine signals with a period equal to the relevant range of date-times (e.g., for yearly seasonality, the sine and cosine span a whole year with a value between -1 and 1 for every time step).
  - One-hot encoding transfers the integer representation of date time to binary classification vectors (e.g., the months of the year can be represented as a vector with zeros and ones, where the one indicates the month that corresponds with the other features).
- **Lag features**, contain past values from time series based on data analysis and domain knowledge (e.g., the three most recent wind speeds or the wind speed 24 hours ago).
- **Window features**, are a summary of multiple values of a variable that can be obtained through either a sliding window or an expanding window (e.g., moving average of the previous three values).
- **Combined variable features**, is the combination of different variables (e.g., combination of wind speed and wind direction into two orthogonal components or the mean of multiple wind speed measurements).

2-2-4 Machine learning models theory

This subsection discusses the theory behind the machine and deep learning forecasting models that have been used for this research. In recent years, leveraging the power of AI technologies and big data have revolutionised various fields, for example, speech recognition, image recognition and natural language processing [25]. Because of the extraordinary results, researchers have started applying the underlying models also for time series forecasting. Even though no machine learning method has been developed for time series specifically, the results are promising [37]. The concept of machine learning models is that the model learns from experience to accomplish a specific task based on its performance. The models that are discussed in more detail are Extreme Gradient Boosting (XGBoost), multilayer perceptron (MLP) and long-short-term memory (LSTM).

XGBoost

XGBoost is a tree-based ensemble method using gradient boosting proposed by Chen and Guestrin [8]. The model is less prone to overfitting than other gradient boosting approaches and known for its speed, parallel computing capabilities, automatic cross-validation and sparsity awareness (i.e., able to handle missing data) [43]. Another advantage of the model is that it is fairly easy to create insight into the feature importance. Based on the principles of game theory, the Shapley Additive Explanations (SHAP) value can explain tree-based learning models. The SHAP indicates to what extend features have been used to generate the predictions of the XGBoost algorithm [31]. The working principle of the model is to minimise the loss function by constructing better trees on a modified version of the original data by following an iterative process. The loss function with regularisation term that penalises model complexity is Eq. (2-6). Traditional optimisation methods cannot solve Eq. (2-6); therefore Eq. (2-7), is generally used in practice to iteratively evaluate split candidates. The three terms within brackets (i.e., often referred to as the gain) are the similarity scores for the left branch, right branch and root leaf. The similarity score is the squared value of the summed residuals of the leaf divided by the number of residuals plus lambda (λ), the regularisation parameter, which is zero by default but can be increased to prevent overfitting. The gamma (γ) represents the tree complexity parameter and influences the depth of the tree through pruning.

$$\begin{aligned} \mathcal{L}(\phi) &= \textstyle\sum_i l\left(\hat{y}_i, y_i\right) + \sum_k \Omega\left(f_k\right) \\ \text{where } \Omega(f) &= \gamma T + \tfrac{1}{2}\lambda \|w\|^2 \end{aligned} \tag{2-6}$$

$$\mathcal{L}_{\text{split}} = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{\left(\sum_{i\in I} g_i\right)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma \tag{2-7}$$

Multilayer perceptron (MLP)

The MLP is the most widely used form of a feed-forward artificial neural network (ANN) and can find the non-linear relationship between inputs and outputs. Feed-forward refers to

the fact that the neurons exclusively propagate their signal forward through the weighted connections between consecutive layers. Moreover, the layers are fully connected, meaning every neuron output is propagated to every neuron in the next layer. The network generally consists of three components: the input layer, one or more hidden layers referred to as the hidden layer and the output layer. Figure 2-3 shows the most basic architecture of a MLP model.

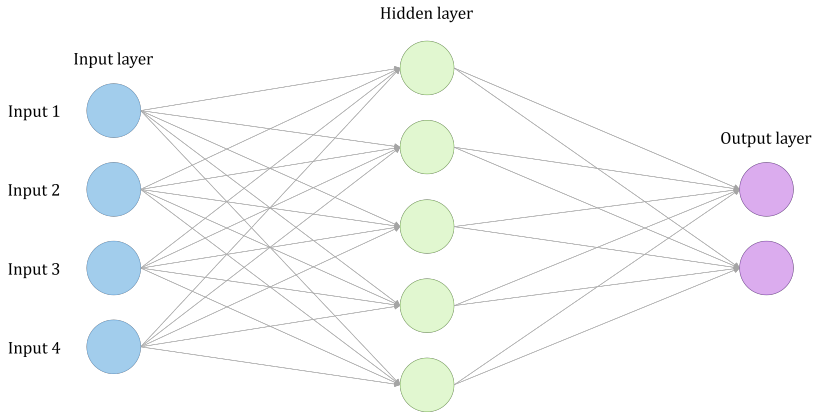


Figure 2-3: Architecture of a basic MLP model.

Except for the input layer, all neurons within a layer have a specified activation function, which realises the non-linear capabilities of MLP. Figure 2-4 visualises the working of a neuron. The inputs of the neuron are multiplied with the weights of the connections (W), and a bias term (b) is added. After which the result (z) is passed through an activation function, which gives either the prediction ($\hat{y}$) or the input for the next layer of neurons (a). The four most common activation functions (σ) are summarised in Table 2-4

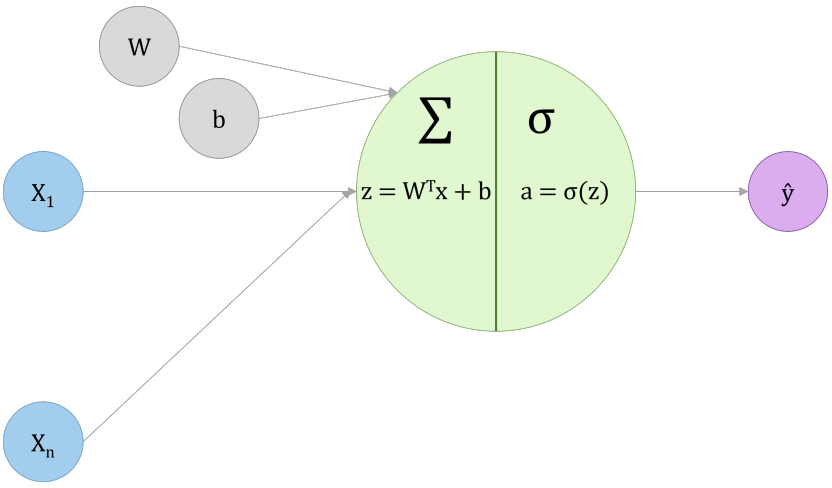


Figure 2-4: Zoom in on the operations within a single neuron with the equations in vector notation.

Table 2-4: Overview of common machine learning activation functions.

Name	Function	Range
Linear	$\sigma(z) = z$	$(-\infty, \infty)$
Hyperbolic tangent	$\sigma(z) = tanh(z)$	$(-1, 1)$
Logistic	$\sigma(z) = \frac{1}{1+e^{-z}}$	$(0, 1)$
Rectified Linear Unit (ReLU)	$\sigma(z) = max\{0, z\}$	$[0, \infty)$

The number of engineered features determines the size of the input layer, while the hidden layer is free for design. For supervised learning (i.e., machine learning where the desired output is known), the output layer architecture is determined by the expected results, referred to as the labelled data. During the training process, the randomly generated weights of the MLP are optimised to minimise the loss function through the backpropagation process, which is fundamentally driven by the partial derivatives of the loss function and the set learning rate. The chosen optimisation algorithm governs this learning process.

The basic steps of the backpropagation learning process are:

1. Forward pass the inputs to predict the output(s).

2. Compare the prediction to the ground truth and compute the loss.

3. The loss function is used for backpropagation through computing the partial derivatives. The gradients are calculated for every node and in combination with the learning rate, influence to what extent the connections' weights are adjusted.

Long short-term memory (LSTM)

The MLP maps the inputs to the outputs but does not capture a dependence between both. This is not a favourable characteristic for data sets where a strong correlation between the inputs and outputs exists, which can be the case for sequential or time-series data. Opposed to the feed-forward ANN the recurrent neural networks (RNN) have loops between layers; therefore, historical data can influence the current prediction. The basic architecture of a RNN is displayed in Figure 2-5. Except for the recursive hidden layers, this network is similar to the MLP; there is an input layer, hidden layer and output layer. The training of this network is also through backpropagation. This causes an issue for RNN that is also observed when training very deep MLP networks (i.e., a network with many hidden layers); namely, the gradients in earlier layers will either vanish or explode. Because the gradients are a product of the gradients of the deeper layers, this results in exponential shrinking or growing of the gradients. Consequently, the model might be unable to converge.

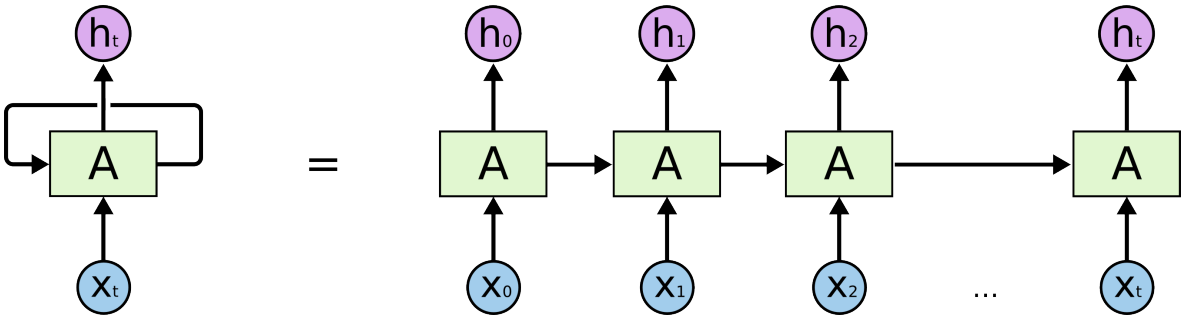


Figure 2-5: Visualisation of an unrolled RNN neuron [38].

Two specific RNN models were invented to solve this problem, namely, LSTM and gated recurrent unit (GRU). These models can learn long-term dependencies through the use of gates, which either remove or conserve information. Therefore part of the error can be directly transmitted to the subsequent network layer. In theory, the error should not disappear even for very long sequences or deep networks. However, in practice, it is necessary to limit the sequence length [43]. In theory LSTM should yield better results than GRU as it has three gates instead of two, which means more parameters to train. For the same reason, GRU is faster to train. In practice, the LSTM and GRU results do differ per data set and problem, but overall show very similar results. Based on the theoretical argument, only LSTM is further explained. In Figure 2-6 the LSTM unit is inserted in the earlier visualised RNN architecture and the three gate components are highlighted. Figure 2-7 provides a more clear visualisation of the separate components of an LSTM unit.

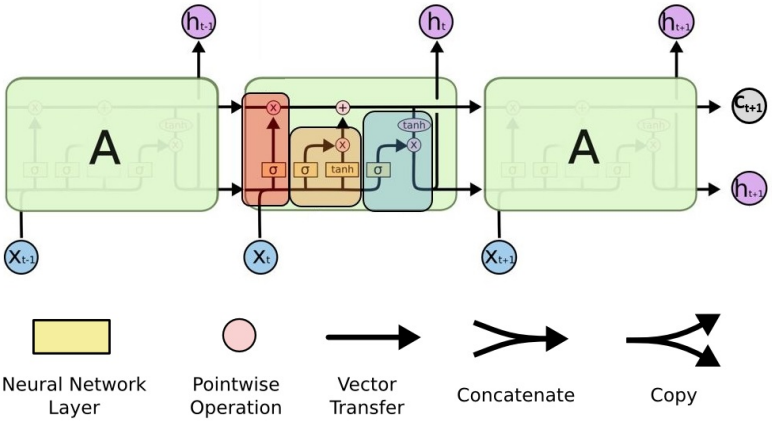


Figure 2-6: Unrolled LSTM block with indication of forget gate (red), input gate (orange) and output gate (blue) [38].

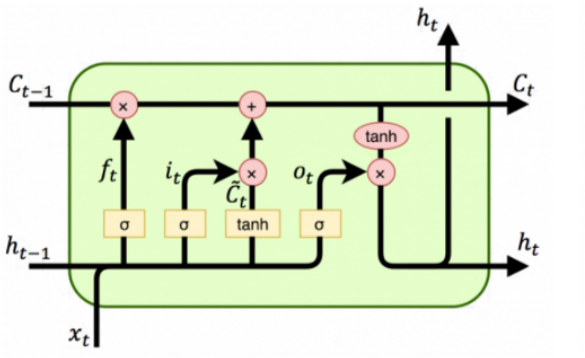


Figure 2-7: Zoom in on computations within one LSTM block [38].

The variable convention used in Figure 2-7 matches the convention in the following equations. The forget gate determines what information of the old cell state is discarded. The forget gate (f_t) takes a value between 0 and 1 where the min and max represent no information or all information is passed forward, respectively. The inputs consist of the previous output of the hidden layer (h_t) and the current input sequence (x_t). These are concatenated and multiplied with the weight matrix (W_f) to which the bias vector (b_f) is added. The whole computation is shown by Eq. (2-8).

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{2-8}$$

The input gate determines to what extent the previous cell state is updated with the preliminary cell state information. The computation of the input gate (i_t) is similar to the forget gate but with different weight matrix (W_i) and bias vector (b_i). The input gate and preliminary cell state information ($\tilde{C}_t$) equations are shown by Eq. (2-9) and Eq. (2-10), respectively.

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{2-9}$$

$$\tilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \tag{2-10}$$

The new cell state combines the retained information of the previous cell state with the filtered preliminary cell state information. This new cell state (C_t) is computed through Eq. (2-11), where ($*$) indicates element-wise multiplication.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{2-11}$$

The output gate filters the hyperbolic tangent of the cell state to construct the hidden layer output Eq. (2-13). Notice that the output gate equation Eq. (2-12) is similar to the forget and input equations.

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \tag{2-12}$$

$$h_t = o_t * \tanh\left(C_t\right) \tag{2-13}$$

2-2-5 Multi-step forecasting styles

The models discussed in subsection 2-2-4 do not necessarily predict multiple steps. However, for some time series forecasting applications, this is required. Therefore, this subsection describes the four main methods for forecasting multiple steps into the future [37].

Direct multi-step forecast strategy

This method makes use of a separate model for each step into the future. The main disadvantage of this method is the loss of dependency on the earlier time steps. Moreover, all the models have to be maintained. There are two particular ways in which this method can be applied.

- *Specific:* The specific method only predicts a specific timestamp. To illustrate, a separate model for each specific hour of the day ahead.
- *Lead time:* The lead time method predicts a specific lead time in the future. Therefore this leads to more optimal use of the available training data.

Recursive multi-step forecast strategy

This method makes a one-step-ahead prediction and appends this prediction to the available inputs to forecast multiple steps into the future (e.g., forecast the value at t+1 with a model and use this prediction to predict t+2 with the same model, this is repeated until the required forecast horizon is reached).

Direct-recursive hybrid multi-step forecast strategies

The direct-recursive hybrid method combines the methods mentioned above: the direct models are used within a recursive framework (e.g., use the prediction from t+1 from model 1 to predict the next time step that is t+2 with model 2). In this manner, both methods can complement each other, but this is inherently more complex.

Multitask forecast strategy

One model is used to forecast a sequence. These models are more complex and often take longer to train. An important consideration when applying the models is whether to weigh the importance of the forecast horizons differently.

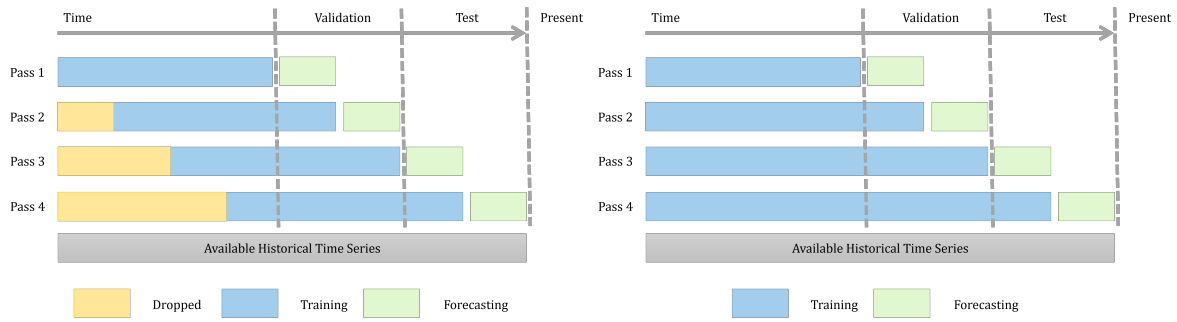
2-2-6 Evaluation

The objective of time series forecasting models is to make accurate predictions on unseen data as this gives the best proxy for future operational performance. The process can be briefly summarized in the following steps. Firstly, determine which data is available and

suitable for the forecasting problem. Secondly, decide on how to resample the data to assess the performance on unseen data. Finally, choose a performance metric that is suitable to evaluate the forecast. The last two steps are extensively discussed in this section.

In machine learning, it is common to resample the data into three categories: training data, validation data, and test data. The training data set has the most samples; based on this set, the model weights are adjusted. The validation data is used for hyperparameter optimisation. The test data shows the performance of the tuned model on unseen data. There are three methods to divide the available historical data while taking temporal dependence into account:

- Split the data into a train, validation and test set (e.g., train on data from the year 2018, validate on data from the year 2019 and test on data from the year 2020).
  - The advantage of this method is that only one model is trained.
- Multiple train, validation and test sets (e.g., train on data from summer 2018, validate on data from summer 2019 and test on data from summer 2020).
  - This requires training multiple models to cover the full range of historical data but can be more accurate if the chosen periods have comparable data characteristics.
- Walk-Forward Validation retrain the model when new data is available through applying a sliding or expanding window as visualised in Figure 2-8.
  - The model is updated after every sliding step; therefore, the most recent data is included, which can improve the forecast. Either the available computational power or the conflict between training time and available forecast time can be a constraint for this method.



(a) Sliding window **(b)** Expanding window

Figure 2-8: Visualisation of the two different types of walk-forward validation.

The evaluation of the forecast performance is referred to as backtesting. The quality of the models can be assessed through different performance metrics. Specifically the standard statistical error metrics root-mean-square error (RMSE) and mean absolute error (MAE), which give an indication of the forecast accuracy occur frequently in research papers [15, 33]. The reason behind their popularity is the fact that these single error metrics are easy to interpret, which makes them useful to compare different models or configurations of models [33]. However, this can lead to the common fallacy that the forecast that performs best on

a specific error metric truly is the best forecast [34]. For this reason, without context, one should always be cautious when interpreting these metrics. Since the focus of this thesis is not on the development of an optimal forecast evaluation framework, the curious reader is referred to the recent paper from Messner et al. [34]. Instead, the following part explains the most frequent standard error metrics [1, 15, 34]. Most of these error metrics can be normalised by dividing the predicted value and the actual value by the maximum observed or theoretical actual value [15].

mean bias error (MBE) Represents the difference between the mean of the predicted values and the mean of the actual values as seen in Eq. (2-14), where N is the number of samples, y_t is the actual value, and $\hat{y}_t$ is the predicted value. A low MBE does not provide insight into the forecast accuracy but should be interpreted as a prerequisite of a good forecast (i.e., an unbiased forecast). If the MBE is significantly positive or negative, this indicates that the model is either over-predicting or under-predicting, respectively.

$$MBE = \frac{1}{N}\sum_{t=1}^{N}(\hat{y}_t - y_t) \tag{2-14}$$

mean absolute error (MAE) Represents the mean absolute difference between the forecasts and expected results. This error metric is suitable for processes with a fixed marginal cost, which means that a larger error does not have to be penalised more than a small error. The MAE is expressed in the original unit, which makes the metric intuitive. The formula for the MAE is given by Eq. (2-15).

$$MAE = \frac{1}{N}\sum_{t=1}^{N}|\hat{y}_t - y_t| \tag{2-15}$$

mean absolute percentage error (MAPE) Represents the mean absolute difference between the forecast and the expected result divided by the expected result. The MAPE is generally reported as a percentage value and therefore considered easy to interpret. However, in the original form, see Eq. (2-16), comes with some issues. The most important one is that the metric becomes unstable when the actual value is zero or close to zero. Additionally, the asymmetric character punishes over-forecasting more compared to under-forecasting. Modified MAPE definitions have been developed to mitigate these issues and inherently more complex.

$$MAPE = \frac{100}{N}\sum_{t=1}^{N}\left|\frac{y_t - \hat{y}_t}{y_t}\right| \tag{2-16}$$

mean squared error (MSE) Represents the mean squared difference between the forecasts and expected results and can be calculated with Eq. (2-17). Because of the quadratic nature of this error metric, more significant errors are penalised heavily. This is the most widely used loss error metric for regression problems.

$$MSE = \frac{1}{N}\sum_{t=1}^{N}(\hat{y}_t - y_t)^2 \tag{2-17}$$

root-mean-square error (RMSE) Is the square root of the MSE, see Eq. (2-18). By taking the square root, the original unit is attained, which makes it easier to interpret.

$$RMSE = \sqrt{MSE} \tag{2-18}$$

2-3 State of the art

The state of the art has been studied and summarised to create insight into the potential of deep learning models for UST-WPF. Specifically focusing on which features, techniques and error metrics were applied, which corresponds with the earlier formulated research questions. An entirely fair comparison between different UST-WPF studies is hard to make due to, for example, the difference in user data sets [30]. Nevertheless, an attempt has been made to create a context in which this thesis fits. Table 2-5 summarises the general information about the most relevant papers that have been studied during the literature review. Figure 2-9 displays the normalized root-mean-square error (NRMSE) results on a 16 programme time unit (PTU) horizon (as defined in the paper of Wu et al. [46]) for all the papers that have either reported this metric directly or have reported the RMSE and the installed nominal capacity. The following paragraphs explain in more detail the relevance, conclusions and limitations of the papers summarised in Table 2-5.

In 2011, Catalão et al. [6] showed the potential of deep learning methods with and without decomposition in improving the UST-WPF compared to traditional statistical models. The implementation of a neural network (NN) model reduced the MAPE on the three-hour horizon by 30% relative to the ARIMA model. Additionally, this study incorporated WT decomposition as a prepossessing technique in combination with a NN; this further improved the MAPE with a 4% reduction relative to the plain NN model. The wind power forecast in this study is on a national level which means the farms are geographically spread; therefore, the results are hard to compare with individual wind farm studies [42].

Liu et al. [29] reported even better results when implementing WT decomposition for multiple deep learning models, namely a backpropagation NN, RNN and LSTM. The RMSE reduction observed on the first PTU ahead prediction due to adding decomposition to the LSTM where 33.01%, 37.63%, 63.80%, 64.90%, and 65.87% compared to DWT-RNN, DWT-BP, LSTM, RNN and BP, respectively. Within the five PTU forecast horizon, the DWT-LSTM outperforms all the other models on any prediction time ahead. What is striking about the results is that the different models without decomposition perform remarkably similarly on all individual prediction steps ahead. This similarity between BP NN and LSTM models is in conflict with the results found by Li et al. [26]. This study finds an average reduction of 21% RMSE on the first four PTUs ahead in favour of the LSTM model.

Recently the combination of more advanced decomposition algorithms and LSTM have shown promising results for the UST-WPF. Han et al. [14] applied the VMD algorithm with a limited

number of modes, reasoning that there is no physical meaning to decomposing the wind power series into more than three modes. The performance of the VMD-LSTM model is significantly better and remains exceptionally consistent over the forecast horizon compared to the WT and BP NN model. Zhou et al. [47] applied EEMD instead of VMD and includes exogenous variables to improve the forecast. The prediction of this model is optimally weighted with the prediction from a Seasonal Autoregressive Integrated Moving Average (SARIMA); according to the author, the SARIMA model can, to some extend, extract the seasonal information from the raw wind power data. The RMSE reduction observed on the first PTU ahead prediction for the proposed hybrid model were 42.12%, 40.38%, 21.39% and 3.15% compared to GRU, LSTM, EEMD-LSTM and PCA-EEMD-LSTM, respectively. It is observed that the difference in performance between GRU and LSTM is slightly favourable towards LSTM. Moreover, the addition of the EEMD algorithm and the exogenous variables have had the most impact on the RMSE score, while the hybrid configuration (i.e., adding the SARIMA model) results in a relatively small improvement.

Nevertheless, the benefit of hybrid combinations of models is a widely covered theme in the forecasting literature. Ju et al. [23] proposed a LightGBM-CNN model that only slightly outperforms the other models included in the study, which are support-vector machine (SVM), deep neural net (DNN), LightGBM and convolutional neural network (CNN). Every model configuration is trained and tested ten times to remove the stochastic element of machine learning models from the comparison. As the error statistics from these runs are averaged before comparison. Wu et al. [46] reported a more evident improvement through the use of a hybrid model. The performance of the proposed CNN-LSTM hybrid model was compared to an individual CNN and LSTM model. The average RMSE over the one-hour forecast horizon decreased by 25.3% and 14%, respectively. The hybrid model's spatial and temporal capability successfully incorporated the input data measured at different turbines throughout the wind farm.

The studies above are all based on modelling without NWP input data. Lu [30] on the other hand, proposed a traditional statistical model; namely, an AR model based on NWP data from the Weather Research and Forecasting model. This study found that the auto-regressive order is in the range of two to four. The relevance and contribution of this paper are that it applies the same model on two distinct data sets, one with steady and one with unsteady wind conditions. These results underline the influence of the used data set on the magnitude of the error metrics. The limitation of this study is that the used data sets are pretty small compared to the other studies.

To summarise, studies that have applied traditional statistical models show that deep learning models outperform these. Furthermore, the contribution of a traditional statistical model within a hybrid model is relatively small compared to alternative approaches. The main alternative approaches are hybrid configurations of different deep learning models, pre-processing with decomposition algorithms or the inclusion of exogenous inputs. The limitations of these studies are that they focus mainly on statistical error metrics. Moreover, most of the studies do not include a persistence model; therefore, no objective benchmark put the model results into perspective. In addition, the full potential of ensemble modelling is not yet explored in these papers. One could think of taking the median of multiple forecasts to have a more robust and most likely better forecast. This is a standard methodology in NWP models [21]. Finally, none of the studies has explored the potential of a multivariate combination model that consists of both a NWP and non-NWP based model.

Table 2-5: Overview of the most relevant UST-WPF state of the art.

Source	Author	Year	Inputs	Pre-processing	Models	Loss	Capacity (MW)	Forecast horizon	Frequency	Error metrics
[6]	Catalão et al.	2011	Power	WT	ARIMA NNWT	Not provided	Not provided	3 hours	15 minutes	MAPE SSE SDE variance
[29]	Liu et al.	2019	Power	Standard scaling DWT	BP NN RNN LSTM DWT-BP NN DWT-RNN DWT-LSTM	MSE	Not provided	1.25 hours	15 minutes	MAE MAPE RMSE
[14]	Han et al.	2019	Power	VMD	BP NN NNWT VMD-LSTM	Not provided	Not provided	4 hours	15 minutes	MAPE RMSE
[23]	Ju et al.	2019	Temperature Fan state Wind speed Motor speed Wind direction Production in previous 5 minutes Pitch angle Power	Minmax scaling Standard scaling	SVM LightGBM DNN CNN CNN-LGBM	Not provided	Not provided	5 minutes	5 minutes	MAE MSE
[26]	Li et al.	2019	Power Wind speed Wind direction Temperature Pressure Humidity	Minmax scaling Wind direction sin and cos	BP NN LSTM	Not provided	174	1 hour	15 minutes	MAE RMSE
[47]	Zhou et al.	2020	Power Wind speed Wind direction Pressure Temperature Humidity	EEMD (Wind power) PCA (Meteo features)	GRU LSTM EEMD-LSTM PCA-EEMD-LSTM SARIMA-PCA-EEMD-LSTM	Not provided	100	15 minutes	15 minutes	MAE MAPE RMSE
[30]	Lu	2020	Wind speed Wind speed NWP	None	AR with WRF mode	Not provided	200	4 hours	15 minutes	APE MAPE RMSE
[46]	Wu et al.	2021	Power Wind speed Wind direction	Minmax scaling	CNN LSTM CNN-LSTM	MSE	49.5	1 hour	5 minutes	MAE MAPE RMSE NRMSE

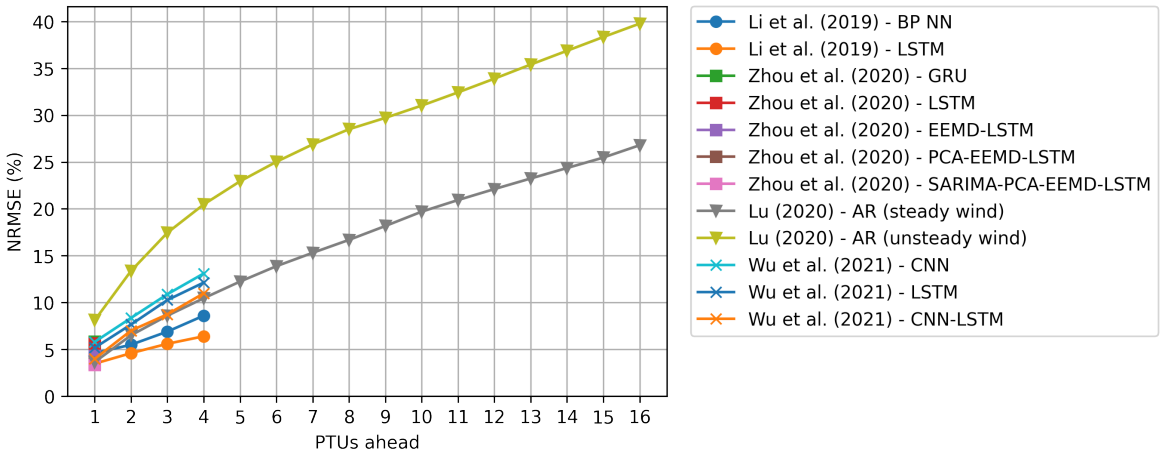


Figure 2-9: Combined results found in literature in one figure for 16 PTUs ahead [26, 30, 46, 47].

Chapter 3

Methodology

The following chapter explains the methodological decisions made during this thesis project. In Chapter 1 the importance of accurate ultra-short-term wind power forecasting (UST-WPF) is explained, and two related research problems are highlighted; application of deep learning methods and forecast value. These topics have been crystallised into two research objectives and five research questions. In chapter 2 the theory about wind power forecasting and time series forecasting is outlined. Additionally, the state of the art on UST-WPF is summarised and discussed. This chapter describes the methodological approach to answer the earlier defined research questions based on the information gathered during the literature review. Section 3-1 describes the data collection and feature engineering process. Next, in Section 3-2 the model is described that was used to gain insight into the importance of the engineered features. Section 3-3 provides details on the proposed forecasting models. Followed by Section 3-4 which lays out the evaluation framework. Finally, Section 3-5 describes the used software and hardware to conduct this research.

3-1 Data collection and feature engineering

This section has been divided into three parts: data collection, data exploration and pre-processing of the data into desirable input features.

3-1-1 Data collection

First, the general information technology (IT) infrastructure is explained. Subsequently, a description of the final data sets used for conducting the research is given. After this, a detailed description of the chronological decision process is given, emphasising the encountered obstacles in data collection.

During the literature review, the importance of real-time data for the UST-WPF was established. Based on interviews with various Eneco employees, the following generic data infrastructure has been mapped and visualised in Figure 3-1.

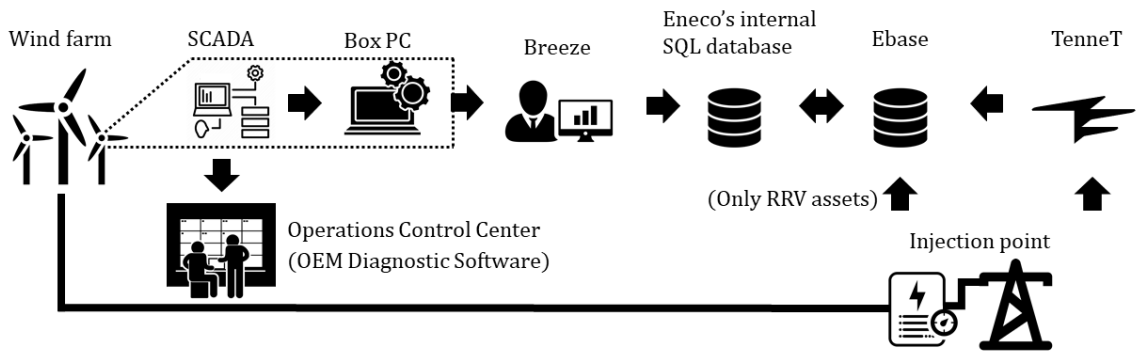


Figure 3-1: Generic overview of data infrastructure at Eneco.

Figure 3-1, depicting the data infrastructure should be interpreted in the following manner. The supervisory control and data acquisition (SCADA) information of a wind turbine is continuously sent to and processed by an internal box PC. The information of every box PC is sent to Eneco's asset management system Breeze (developed by Greenbyte). The Breeze Application Programming Interface (API) is used for a scheduled data drop of the wind speed, wind direction and power on farm level every minute to Eneco's internal Structured Query Language (SQL) database. The bottom data stream is the direct power measurement at the injection point. This signal is continuously reported for the wind farms that are used for regel- en reservevermogen (RRV) (i.e., the Dutch transmission system operator (TSO) term for regulating reserve) and is stored in Ebase (i.e., an energy data management system used at Eneco). TenneT monitors the power output of all wind farms at their respective injection points, which is referred to as the allocation data. However, the allocation data is generally sent with a one day delay to Eneco; therefore, it can only be used as ground truth for model evaluation and not for UST-WPF.

The research is conducted on Princess Amalia Wind Farm (PAWP) which has a nominal capacity of 120 MW and has been operational since 2008. The historical data used for this research ranges from 01/03/2018 23:00 (the date on which Eneco implemented Breeze) until 26/01/2021 22:45 Coordinated Universal Time (UTC). The different data types are summarised in Table 3-1. Meteorological and system availability data were gathered through the Breeze API. Missing meteorological values were imputed from the nearest wind farm, which is Luchterduinen Wind Farm (LUD). Missing system availability values were imputed by dividing the power observation by a theoretical power calculated based on a constructed power curve and available wind speed data. The power and allocation data measured at the grid injection point were exported from Ebase. The remaining missing values were imputed through forward filling. For this particular data set, the maximum number of consecutive Not a Number (NaN) values was nine. Before imputation, all anomalous observations found through their respective theoretical minimum and maximum limit were replaced by NaN values. All time series were resampled into a train, validation and test set as summarised in Table 3-2.

Table 3-1: Overview of the gathered data from Eneco.

Data	Unit	Source	Real-time	Missing values
Wind speed	m/s	Breeze	Yes	Yes
Wind direction	deg	Breeze	Yes	Yes
System availability	-	Breeze	Yes	Yes
Power	kW	Ebase	Yes	No
Allocation	kW	Ebase	No	No

Table 3-2: Chosen resampling of the gathered data expressed in UTC.

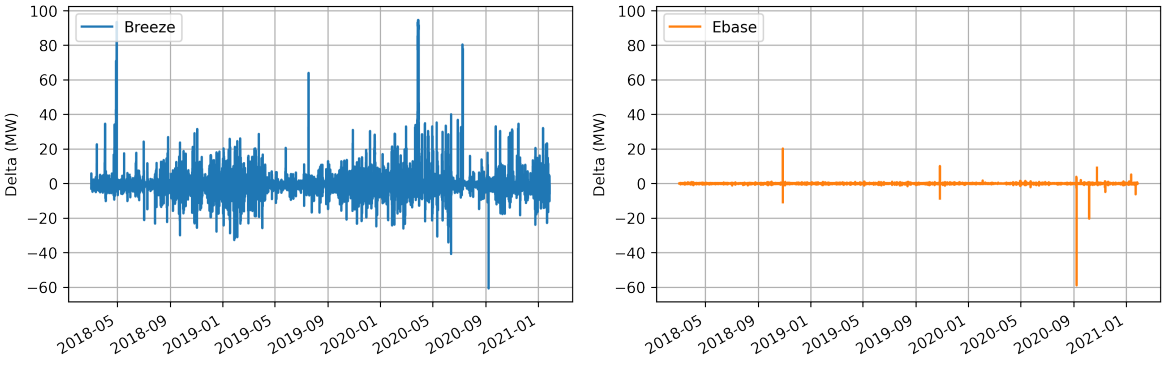
Data set	Start	End	Percentage of data
Train	01/03/2018 23:00	28/06/2020 13:00	80%
Validation	28/06/2020 13:15	31/10/2020 22:45	12%
Test	31/10/2020 23:00	26/01/2021 22:45	8%

The following paragraphs are dedicated to the chronological data gathering process to support and explain the choice of data used for this research project. With the initial intention to develop a generic model based on the current IT infrastructure, the historical wind speed, wind direction and power were queried from Eneco's internal SQL database. Eneco owns wind farms in the Netherlands, Belgium and the United Kingdom. For three reasons, only the Dutch wind farms were considered for this thesis. Firstly, to limit the amount of data. Secondly, the majority of the installed capacity is in the Netherlands. Thirdly, the limited geographical separation between assets makes accurate forecasting more critical and thus commercially attractive. Not all Dutch wind farms report real-time data, which is a requirement for this project. Based on these criteria, the data from the wind farms in Table A-1 was retrieved from Eneco's internal database. After initial data exploration, it was observed that the percentage of missing power values for these wind farms exceeded the required quality due to the contribution of both missing timestamps and missing observations. The percentage of missing values ranged from 3.74% to 21.86% with a median of 13% and an average of 12.68%. The conclusion was drawn that this data flow is too unreliable to use, and imputation would likely cause inaccurate data points. This raised the question wherein the data flow this loss of data had occurred.

It was decided to go one step back in the data flow. Thus a new data request was made through the Breeze API. This opened up the opportunity to add more SCADA data parameters to the query. Based on data exploration in Ebase, conversations with industry experts and intuition, it was decided to add system availability to the query. Even though, to the best of the author's knowledge, system availability has not been incorporated in earlier research on UST-WPF. Not adding additional meteorological features is mainly to restrain the input dependencies of the model, which makes the model more suitable for an operational environment. The new data had a considerably lower percentage of missing power values. This brought to light that the connection between Breeze and Eneco's internal database has not always been stable. The data quality was high enough to proceed to data wrangling. However, after plotting the power data from Breeze and allocation data from Ebase, a significant mismatch

was observed, especially during ramp up and ramp down events. Due to magnitude and bidirectional character, the mismatch could not be ascribed to line losses. This is visualised through plotting the difference between both signals for PAWP in Figure 3-2a.

Multiple steps were taken to deduce the cause of the observed mismatch. The first step was to shift the data to reveal a potential consistent delay or a time zone related issue, but this did not reduce the mismatch. Secondly, the construction of the signal was examined. The Breeze power data is on turbine level and aggregated on farm level per minute. The downsampling from minute to programme time unit (PTU) frequency data happens through taking the exclusive mean. One hypothesis is that since the real-time power signal is generated in this manner, the granularity of the per-minute data might not be high enough to capture the ramp up and ramp down events accurately. The final and most likely cause for the mismatch is the delay and system faults in receiving data from the box PC. Since improving this issue is considered outside the scope of this thesis, the decision was made to focus on building a model for RRV assets only with a real-time power signal at the injection point. The quality of this power signal is visualised through plotting the difference between allocation and the historical real-time injection point power data from Ebase in Figure 3-2b. It can be observed that there is almost a perfect match with the allocation data. The significant spikes that remain have been investigated manually, but no apparent underlying cause was discovered. In order to prevent lookahead, no modifications were made.



(a) Breeze real-time power data

(b) Ebase real-time power data

Figure 3-2: The difference between the historical real-time power data from two different sources and the allocation data for PAWP.

3-1-2 Data exploration

Seasonality

Based on the literature, both a diurnal cycle and yearly seasonality were expected. During data exploration, it became clear that the diurnal effect is negligible compared to the strong yearly seasonality. The yearly seasonality is observed from the monthly mean wind power production of PAWP taken over the whole duration of the data set. This is visualised in Figure 3-3.

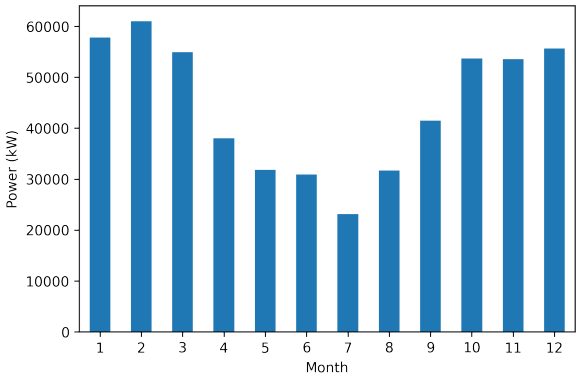
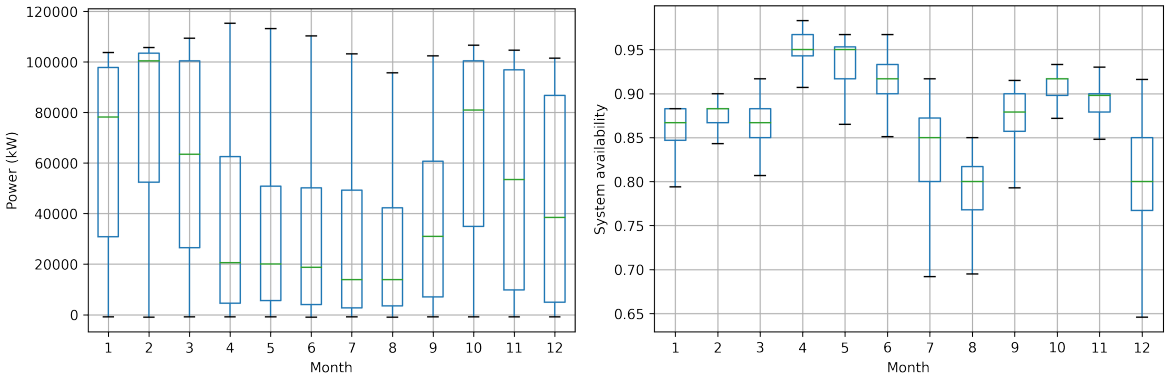


Figure 3-3: Mean allocation per month for the full range of the PAWP data set.

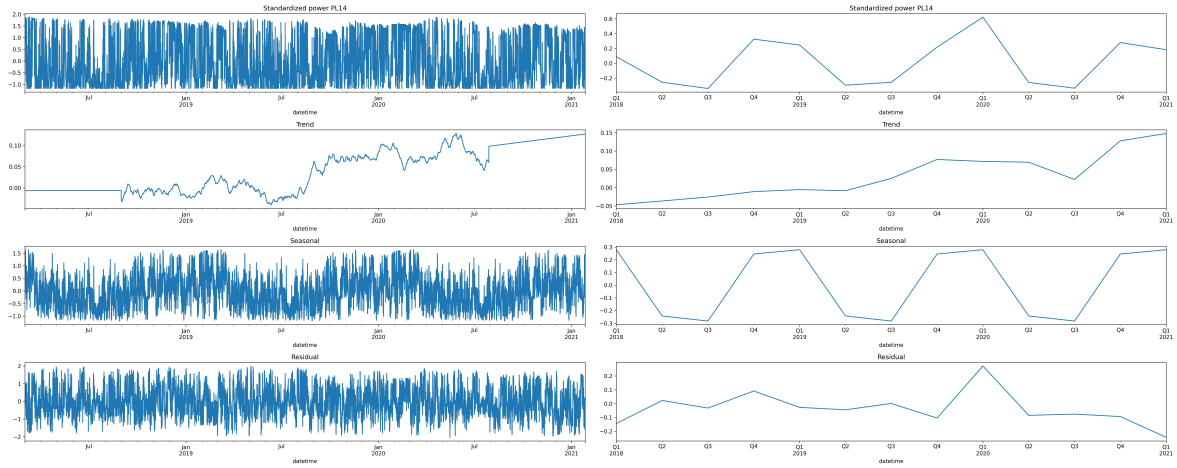
In order to gain more insight into the distribution of the data, the box and whisker plot for the monthly power data of PAWP in 2020 was plotted in Figure 3-4. It can be observed that in every month, approximately the full range of power values occur. However, the median generally follows a similar pattern to the mean values of the previous figure. Moreover, the influence of system availability on power production is best observed for December 2020, which supports the importance of including the system availability data.



(a) Power **(b)** System availability

Figure 3-4: The influence of system availability on the allocation for the year 2020.

The yearly seasonality is also observed through decomposition as described in section 2-2-3. Figure 3-5 shows for both PTU and quarterly frequency the additive decomposition for a yearly periodicity. Because the dataset length is two months short of three full years, the extrapolate trend function has been applied.

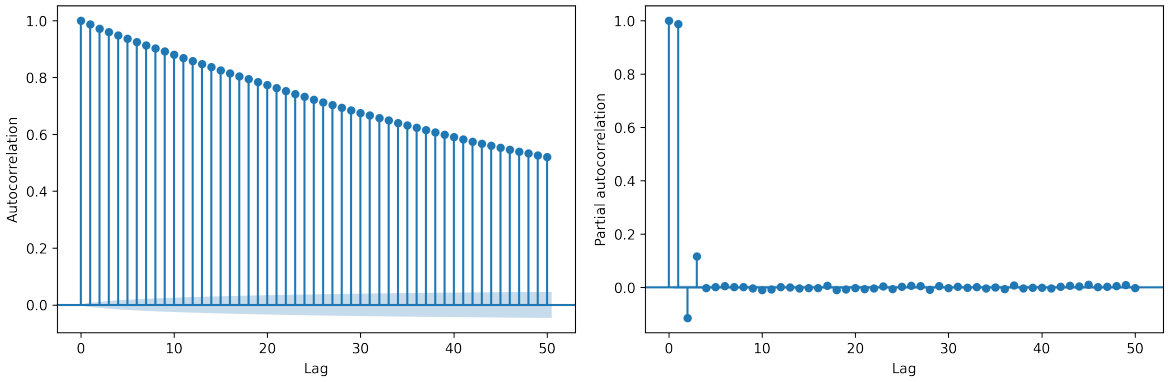


(a) PTU seasonal decompose **(b)** Quaterly seasonal decompose

Figure 3-5: Automatic seasonal decomposition for the allocation data of PAWP.

Correlation

The autocorrelation function (ACF) and partial autocorrelation function (PACF) have been plotted in Figure 3-6 to gain insight in the correlation of the power data set. It is observed that the PACF drastically decreases, and after three lags, no statistically significant correlation is observed. This is in agreement with results from previous studies [30].



(a) ACF **(b)** PACF

Figure 3-6: ACF and PACF for the allocation data of PAWP.

3-1-3 Pre-processing

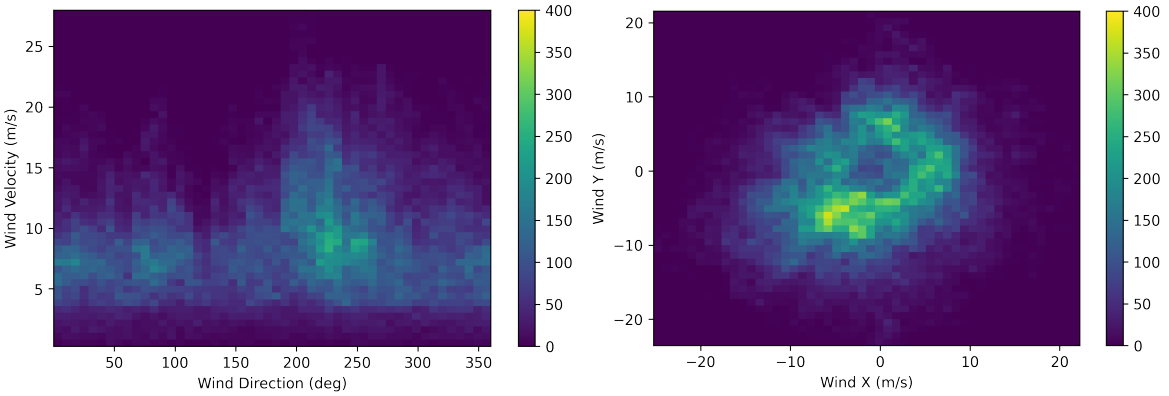
Feature engineering

For this research, three new inputs were engineered; two combined variables and two time-based features. The combined variable features are created because raw wind direction data in degrees does not form a good input feature. There are two reasons for this: the small and

large values are numerically remote while physically distant (e.g., 1 degree and 359 degrees). Secondly, the wind direction measurement does not contain relevant information when the wind speed magnitude is close to zero. For these reasons, the wind direction in radians (θ) and wind speed (v) have been combined into two vectors according to Eq. (3-1) and Eq. (3-2). Consequently, the data distribution has improved, the before and after results are displayed in Figure 3-7.

$$v_x = v \cdot \cos(\theta) \tag{3-1}$$

$$v_y = v \cdot \sin(\theta) \tag{3-2}$$



(a) Distribution of the wind speed and wind direction data **(b)** Distribution of the wind x and wind y vectors

Figure 3-7: The data distribution before and after feature engineering the wind speed and wind direction.

The other two time-based metadata features were constructed with the intention to provide the model with information about the observed yearly seasonality. A yearly sine and cosine were constructed with the timestamp information (t) and the total number of PTUs in one year (T), see Eq. (3-3) and Eq. (3-4). The two features have been visualised for the first year in Figure 3-8.

$$Y_{sine} = \sin(\frac{2\pi \cdot t}{T}) \tag{3-3}$$

$$Y_{cosine} = \cos(\frac{2\pi \cdot t}{T}) \tag{3-4}$$

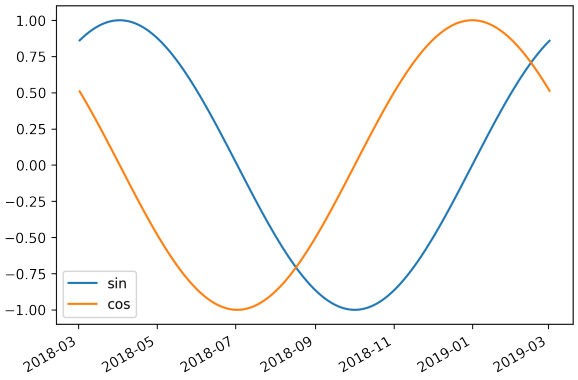


Figure 3-8: Plot of the yearly seasonality features for the first year of the data set.

After constructing these four additional features and removing the wind direction from the available input data, the wind power and wind speed features were transformed using a standard scaler. The standard scaler used the mean and standard deviation of the training data set to prevent lookahead.

3-2 Feature importance model

The Extreme Gradient Boosting (XGBoost) regressor model has been trained with the multivariate inputs as illustrated in Figure 3-9 [8]. The objective of implementing this model was to create insight into feature importance; therefore, the model hyperparameters have not been optimised. Nevertheless, it was checked whether the model outperformed persistence on the first PTU ahead with the settings as summarised in Table 3-3 because otherwise, insights in the feature contributions should be considered meaningless.

Table 3-3: The hyperparameter settings of the XGBoost model.

Hyperparameters	Values
Ensembles	5
n_estimators	1000
max_depth	5
learning_rate	0.02

3-3 Forecasting models

3-3-1 Multilayer perceptron (MLP)

Different multilayer perceptron (MLP) architectures have been applied to prove the hypotheses that the persistence model can be improved, and additional features improve the forecast. There are three reasons to choose MLP as a first model. Firstly, MLP has proven to outperform less complex models (e.g., ARIMA) [6]. Secondly, the neural network is relatively

straightforward to implement and understand. Thirdly the MLP is computationally less expensive than long-short-term memory (LSTM), which makes early-stage experiments less time-consuming. In total, three model architectures were chosen, which have been summarised in Table 3-4. One of the architectures is visualised in Figure 3-10. The performance of three models was explored for both univariate and multivariate inputs see Figure 3-9, which results in a total of six experiments. Based on the findings in subsection 3-1-2 and the literature review, the maximum amount of lag values was set to four. The model was evaluated on the one step ahead forecast. In order to obtain a multi-step ahead forecast, the final MLP model was applied recursively. The values in the final output vector were clipped to the minimum and maximum power values observed in the training set.

For every experiment, a random search numerical optimisation algorithm with 50 iterations was executed to compare only the best performing architecture for each experiment. The Adam optimiser was used to minimise the mean squared error (MSE) loss function [24]. The weights were initialised according to He uniform initialisation [18]. Learning rates were varied during the random search between 0.01 and 0.00001 with a logarithmic step size. The batch size was set to 256 after trial runs with varying batch sizes. The maximum number of epochs was set to 1000, but an early stopping callback with a patience of 5 was included to prevent overfitting the training set. Because when the validation loss does not improve within five epochs, the training stops and the best model is saved.

Table 3-4: Details on the three different MLP random search configurations.

Layer (type)	Activation	Parameters	Model 1	Model 2	Model 3
Dense	Relu	5-100 (step size 5)	x	x	x
Dropout		0-0,5 (step size 0,1)		x	
Dense	Relu	0-100 (step size 5)		x	x
Dense	Relu	0-100 (step size 5)			x*
Dense	Linear	#Outputs	x	x	x

* Randomly included or excluded during random search

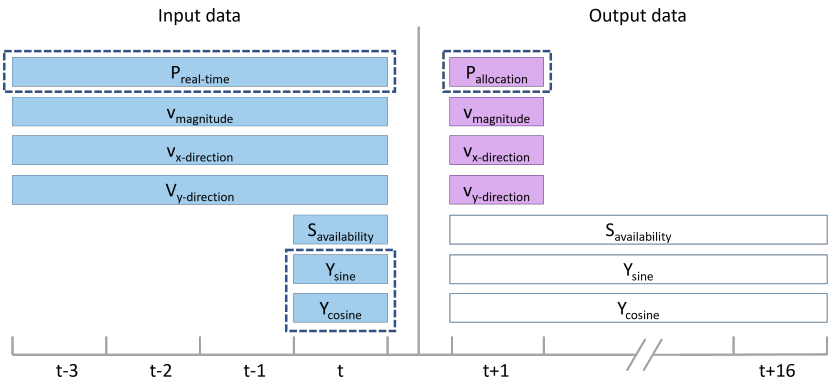


Figure 3-9: Visualisation of the MLP model input features (blue) and output features (pink). The univariate inputs and outputs are indicated through the dashed boxes. The empty boxes with a blue outline represent available future information.

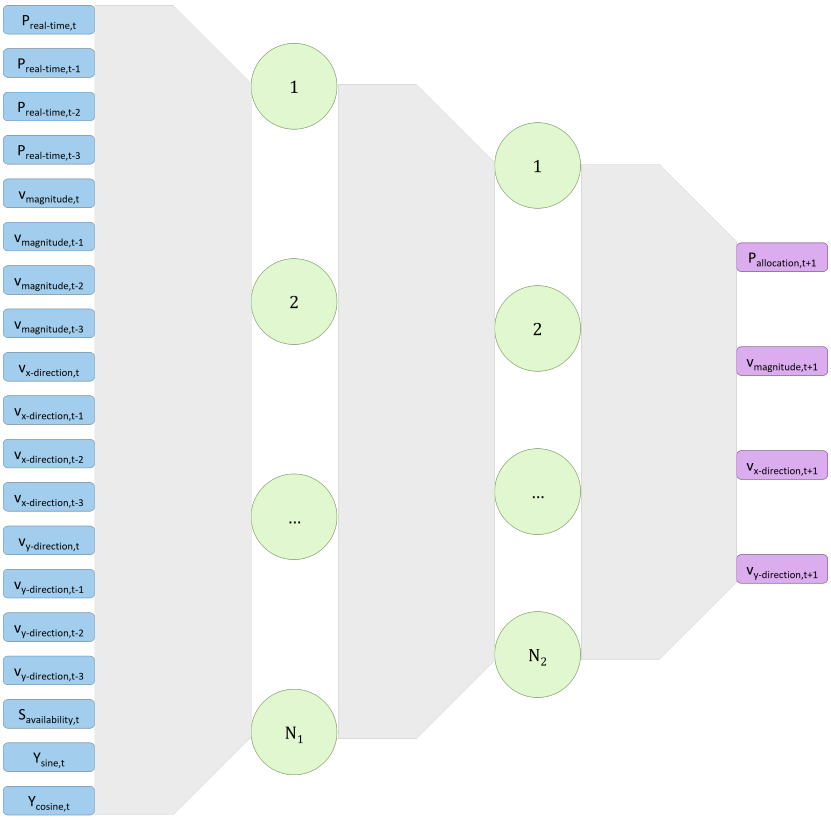


Figure 3-10: Visualisation of the MLP model 3 architecture with two hidden layers.

3-3-2 Long short-term memory (LSTM)

In total two LSTM model architectures were chosen, which have been summarised in Table 3-5. The first model is a Vanilla LSTM and contains a single layer of LSTM units. The second model is a more complex stacked LSTM model, which contains two hidden layers. For this model, the first layer is slightly different as it produces an output for every time step and passes this information to the next LSTM layer. The performance of the two models was only explored for multivariate inputs (see Figure 3-11), which results in a total of two experiments. The model was evaluated on the one step ahead forecast. In order to obtain a multi-step ahead forecast, the final LSTM model was applied recursively. The values in the final output vector were clipped to the minimum and maximum power values observed in the training set.

For every experiment, a random search numerical optimisation algorithm with 50 iterations was executed to compare only the best performing architecture for each experiment. The Adam optimiser was used to minimise the MSE loss function [24]. The weights were initialised according to He uniform initialisation [18]. Learning rates were varied during the random search between 0.01 and 0.00001 with a logarithmic step size. The batch size was set to 256 after trial runs with varying batch sizes. The maximum number of epochs was set to 1000, but an early stopping callback with a patience of 5 was included to prevent overfitting the training set.

Table 3-5: Details on the two different LSTM random search configurations.

Layer (type)	Activation	Parameters	Model 1	Model 2
LSTM	Sigmoid and tanh	8-128 (step size 8)	x	x
LSTM	Sigmoid and tanh	8-128 (step size 8)		x
Dense	Linear	#Outputs	x	x

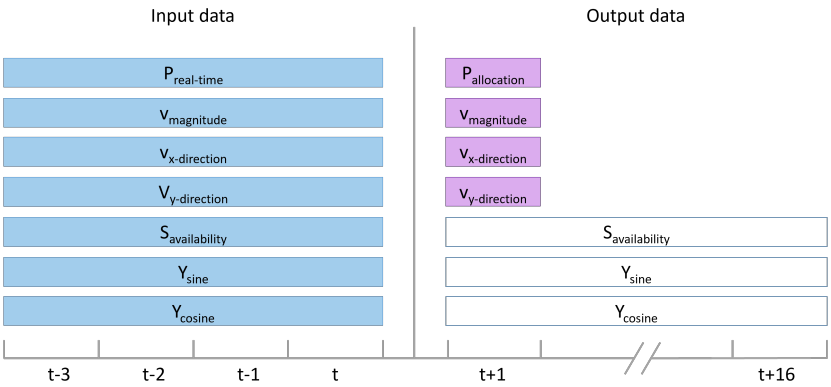


Figure 3-11: Visualisation of the LSTM model input features (blue) and output features (pink). The empty boxes with a blue outline represent available future information.

3-3-3 Ensemble models

In order to create insight into the stochastic nature of the models and make the final forecast more robust, ensemble models were produced. After the hyperparameter optimisation for both the MLP and LSTM model, ten ensemble models were trained. For every trained model, the weight initialisation was slightly different, which results in different models and consequently results. Subsequently, for every prediction from the ten models, the median, the 10th percentile, and the 90th percentile were stored, see Figure 3-12. The latter two are used to quantify the stochastic nature of the models. The median of the ensembles is proposed as the final forecast. This methodology is comparable to the filtering approach, which is often applied to numerical weather prediction (NWP) based models [21].

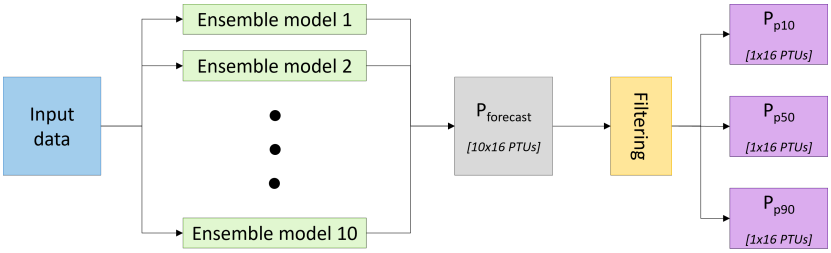


Figure 3-12: Illustration of generating the percentile forecasts.

3-3-4 Multivariate combination methods

Another technique to further improve the forecast was to equally weigh the Eneco model predictions with the median predictions of the MLP models and the LSTM models. This method is known as the multivariate combination method [21].

3-4 Evaluation setup

In this section, the model evaluation process is explained for both forecast quality and value. Subsequently, the details about the two benchmark models are described.

3-4-1 Evaluation

The model modification and selection process were based on the MSE loss results on the validation set for the one-step-ahead prediction. In the case of multivariate outputs, all outputs were equally weighed.

In order to simulate the operational implementation and thus the true performance of the best models, a sliding window was moved over the test set with a step size of 1 PTU; after each iteration, the next 16 steps were recursively forecasted and recorded. The forecasted values were used to calculate both the forecast quality and forecast value for the relevant lead times.

Forecast quality metric

The accuracy of the models is evaluated with three different standard error metrics; namely, the root-mean-square error (RMSE), the mean absolute error (MAE) and the mean bias error (MBE). The primary metric for comparing the accuracy of the models is the RMSE. Because of following reasons:

- As described in the section 2-2-6 it is a common wind power forecasting (WPF) error metric, which enables the possibility to compare the results with work from other researchers.
- The error metric is expressed in the original unit, which makes it intuitive to interpret.
- The RMSE is a single error metric, which makes it easy to compute and understand.

The reason behind including the MAE is to check whether this evaluation metric supports the results of the RMSE metric. The MBE on the other hand is to validate whether the model is unbiased.

Forecast value metric

In this subsection, first, a brief overview of the electricity market process is given. Subsequently, it is explained how this process is adjusted and which assumptions were made to conduct a static value analysis. Finally, a step by step overview of the applied framework with the relevant equations is given.

As explained in Chapter 1 a Balancing Responsible Party (BRP) has to report the expected wind power production per hour for the next day at noon the day before delivery, which is sold in the day-ahead market.

During the delivery day, the intraday trader aims to trade towards the adjusted forecast for every PTU of the day on the intraday market. The intraday market has two closing times, one for European Cross-Border Intraday (XBID) and one for national trading, see Figure 3-13. On the intraday market, most trading contracts are hourly, but half-hourly and quarter-hourly contracts exist. If the intraday trader wants to trade hourly contracts, at least a five and nine PTU ahead forecast horizon is required for the national and XBID market, respectively. From here onward, the five and nine PTU forecast horizons are referred to as the national and XBID horizon. Figure B-1 illustrates the necessity of a nine PTU ahead forecast horizon to trade within the XBID market. The significant advantage of trading in the XBID market is that there is generally more liquidity than in the national market.

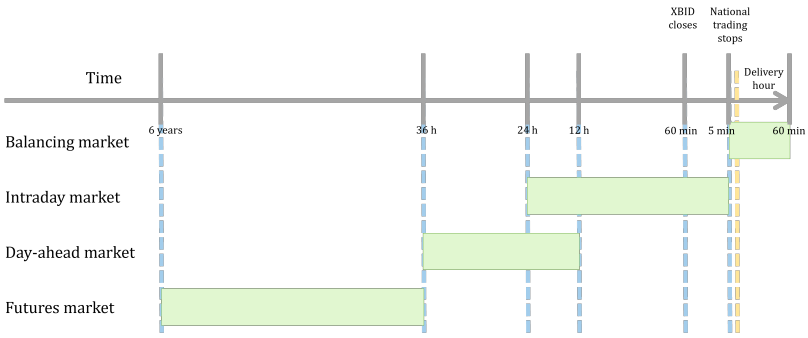


Figure 3-13: The timeline of the different electricity markets.

The difference between the obtained position for every PTU and the allocation per PTU (i.e., the actual delivery) is settled against the corresponding imbalance price per PTU. Generally, there is one imbalance price per PTU. However, the imbalance price can sometimes differ between upward regulation (i.e., take price) and downward regulation (i.e., feed-in price); this is referred to as 2-sided regulation and is due to steering within the timeframe of one PTU.

The static value analysis presented in this thesis aims to measure the financial performance of a proposed model relative to the Eneco model. Since the metric is relative to the Eneco model, the reported day ahead position is not required. The most important assumptions made for the analysis are:

- The last hourly mean of the Eneco forecast is obtained for every delivery hour.
- The trades and positions do not influence the intraday and imbalance price.

- Only 50 per cent of the total traded capacity at the specific lead time of a delivery hour is available to adjust the intraday position.
- Only hourly contracts are traded.
- The trader does not have a strategic mindset.

The framework computes the financial performance ($\mathbf{Tval_{tot,diff}}$) for every PTU in the test set. The framework consists of two parts. The first four steps explain how the intraday and imbalance volumes were calculated. The last steps explain the financial settlements of these volumes.

1. The mean forecast of the proposed model ($\overline{P_{P,n,i}}$) and Eneco logging ($\overline{P_{E,n,i}}$) is calculated for all lead times (N) indicated with subscript (i) on all tradable delivery hours (M) indicated with subscript (n), where N depends on the forecast horizon and M on the length of the test set.

2. The tradable intraday volume ($\Delta_{id,n,i}$) with respect to the Eneco model is computed through Eq. (3-5). The total available volume ($Q_{sum,n,i}$) at the specific lead time before delivery is multiplied with the assumed available market for Eneco (ω), which is set to 0.5. This signifies that 50% of the trade volume is available for Eneco to change position. The model trades in chronological order and account for earlier position changes. To prevent making the same correction multiple times.

$$\Delta_{id,n,i} = \begin{cases} \overline{P_{P,n,i}} - \overline{P_{E,n,i}}, & \text{if } |\overline{P_{P,n,i}} - \overline{P_{E,n,i}}| \leq \omega \cdot Q_{sum,n,i} \\ sign(\overline{P_{P,n,i}} - \overline{P_{E,n,i}}) \cdot \omega \cdot Q_{sum,n,i}, & \text{else} \end{cases} \tag{3-5}$$

3. The post-intraday position of the proposed model ($\mathbf{P_{P,p\text{-}id}}$) is computed through adjusting the last forecasted mean Eneco position ($\overline{\mathbf{P_{E,i=1}}}$) with the cumulative intraday volume over all tradable lead times. The post-intraday position of the Eneco model ($\mathbf{P_{E,p\text{-}id}}$) is equal to $\overline{\mathbf{P_{E,i=1}}}$ following the model assumptions. Where the hourly values are upsampled to quaterly values.

$$\mathbf{P_{P,p\text{-}id}} = \overline{\mathbf{P_{E,i=1}}} + \sum_{i=1}^{N} \Delta_{id,n,i} \tag{3-6}$$

4. For both the proposed model and Eneco model, the imbalance volume for every delivery PTU is calculated through subtracting the post intraday position from the allocation data ($\mathbf{P_A}$), resulting in ($\mathbf{\Delta_{imb,P}}$) and ($\mathbf{\Delta_{imb,E}}$), respectively.

5. The tradable intraday volume is traded against the volume weighted average price (VWAP) for the forecasted hour at the relevant trade time interval ($€_{id,n,i}$).

6. The imbalance volume vectors ($\mathbf{\Delta_{imb,P}}$) and ($\mathbf{\Delta_{imb,E}}$) are traded against their respective imbalance price vectors ($€_{\mathbf{imb,P}}$) and ($€_{\mathbf{imb,E}}$). These price vectors are calculated according to Eq. (3-7).

$$€_{imb,n} = \begin{cases} €_{take,n}, & \text{if } sign(\Delta_{imb,n}) = -1 \\ €_{feed\text{-}in,n}, & \text{else} \end{cases} \tag{3-7}$$

7. The financial performance vector of the proposed model can be determined by Eq. (3-8), where ($\mathbf{Tval_{tot,P}}$) represents the total trade value of the proposed model as defined by Eq. (3-9) and ($\mathbf{Tval_{tot,E}}$) the total trade value of the Eneco model as defined by Eq. (3-10). The summed intraday trade value is filled for every PTU in the delivery hour to match the length of the imbalance volume and price vector.

$$\mathbf{Tval_{tot,diff}} = \mathbf{Tval_{tot,P}} - \mathbf{Tval_{tot,E}} \tag{3-8}$$

$$\mathbf{Tval_{tot,P}} = 0.25 \cdot \sum_{i=1}^{N} (€_{id,n,i} \cdot \Delta_{id,n,i}) + 0.25 \cdot \mathbf{€_{imb,P}} \cdot \mathbf{\Delta_{imb,P}} \tag{3-9}$$

$$\mathbf{Tval_{tot,E}} = 0.25 \cdot \mathbf{€_{imb,E}} \cdot \mathbf{\Delta_{imb,E}} \tag{3-10}$$

3-4-2 Benchmark models

Persistence

The persistence method is applied to create a meaningful benchmark to check whether a model has skill. Another advantage of including the persistence method is to provide future researchers with an indicative measure of how predictable this specific test set was. Through computing percentage improvements to persistence, this study can be easily compared with other works.

Eneco model

Out of commercial interest, it is important to gain insight into whether the proposed models outperform the companies status quo model. Detailed information about this model has to remain confidential. Nevertheless, what can be shared is the general working principle of the Eneco model. The Eneco model is a k-nearest neighbour NWP based model with a two-step delayed rudimentary ultra-short-term (UST) correction model based on real-time data. If for two subsequent PTUs the deltas between the real-time signal and the forecast have the same sign. The forecast is corrected with the last delta multiplied by a factor that decays with respect to the forecast horizon. The 0, 4, 6, 8, 12, and 16 PTU ahead logging of the UST corrected Eneco model was used. It is assumed that these loggings can be linearly interpolated for analysis. These loggings are available from November 2020 onward and thus constrained the duration of the test set.

3-5 Software and hardware implementation

3-5-1 Software

The code has been written in the programming language Python 3, using Jupyterlab Integrated Development Environment (IDE) from package management software Anaconda. The following Python packages were used to develop the models:

- Keras 2.4.3
- Tensorflow 2.3.0
- scikit-learn 0.24.1
- XGBoost 1.3.3

Neptune.ai 0.5.1 was used to keep track of the different experiments.

3-5-2 Hardware

Developing the models and trial runs were executed on Eneco's Data Science Virtual Machine (DSVM), which is a 24/7 operational Azure D4s v3 virtual machine. It contains an Intel(R) Xeon(R) CPU E5-2673 v4 processor and 16GB random-access memory (RAM). The operative system is Windows Server 2016 Datacenter 64-bit.

Some experiments were executed on the Azure NC6s_v2, which is a Deep Learning Virtual Machine (DLVM) (i.e., a type of DSVM designed for deep learning applications) [41]. The main difference with Eneco's DSVM is that the DLVM contains a Graphical Processing Unit (GPU), specifically the NVIDIA Tesla P100, which significantly reduced the training time of the models. Moreover, the DLVM contains an Intel(R) Xeon(R) E5-2690 v4 processor. It runs on the same operating system as the DSVM.

Chapter 4

Results & Discussion

This chapter displays the results and discusses their relevance. The first section focuses on the results from the feature importance model. The following section summarises the hyper-parameter optimisation results for the two proposed deep learning models. Subsequently, in section 4-3 the forecast quality is evaluated, followed by the results of the valuation model. The final section presents six case studies to create insight into the forecast behaviour under specific circumstances.

4-1 Feature importance model

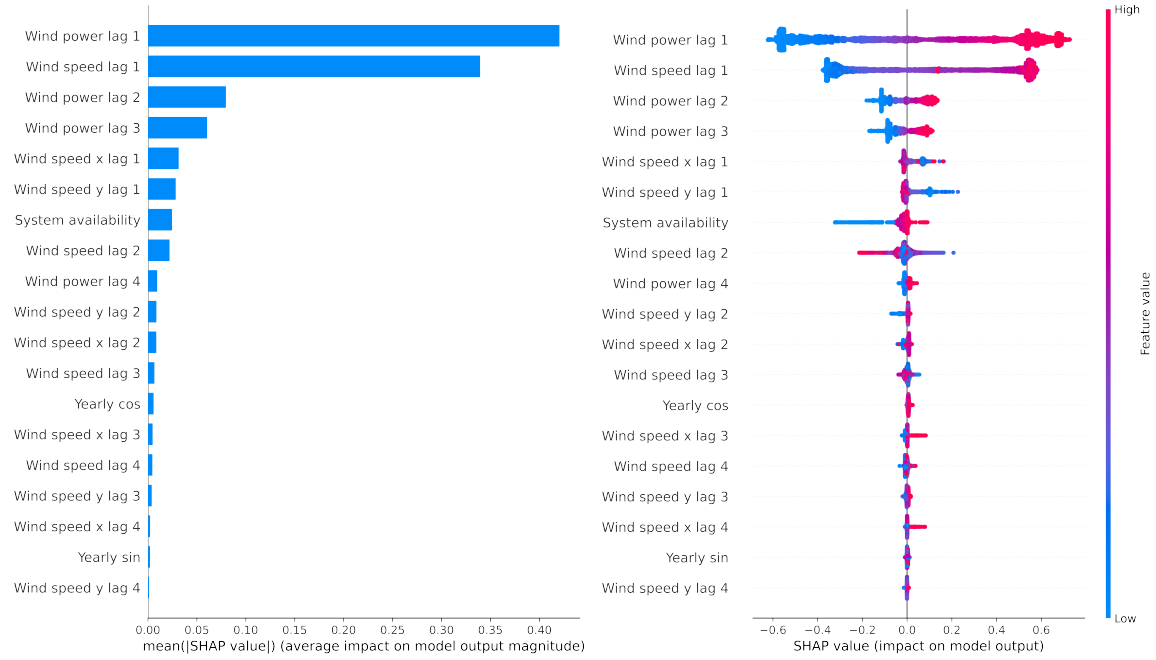
One of the research questions was to investigate what set of input variables are most relevant on the ultra-short-term (UST) time scale. Even though this questions is to some extent already answered in subsection 2-1-1 and section 2-3. Out of curiosity, it was decided to implement an Extreme Gradient Boosting (XGBoost) model to gain more insight into the used features through the Shapley Additive Explanations (SHAP) methodology. As the SHAP methodology used, is not suitable in its original form for explaining deep neural net (DNN) time series models [27]. These findings cannot be extrapolated to the forecasting models but still deliver interesting insights.

Table 4-1 shows the root-mean-square error (RMSE) scores of the XGBoost model on the one-step-ahead forecast horizon for the training, validation and test set. The model's performance on the test set can be compared with the data in Table C-1 which shows that the model has skill compared to the persistence model. Consequently, the feature importance results likely contain meaningful information.

Table 4-1: XGBoost results for one-step-ahead forecasting.

Run	Training RMSE (kW)	Validation RMSE (kW)	Testing RMSE (kW)
Ensemble 1	4755	5419	4920
Ensemble 2	4691	5410	4853
Ensemble 3	4586	5404	4834
Ensemble 4	4661	5333	4852
Ensemble 5	4611	5337	4854
Average	4661	5381	4863

What stands out in Figure 4-1a is the importance of the most recent wind power and speed, while there is strongly decreasing importance of more distant lagged values. These results are in alignment with the findings in the literature review and data exploration section. Another important result is the significant impact of the system availability. Interestingly, in Figure 4-1b the system availability shows to have the most influence when the feature value is low. These findings suggest the importance of system availability for low wind power predictions. The two time-based features that should capture the yearly seasonality seem to have a minor influence on the forecast results. It is difficult to explain this disappointing result, but it might be related to how the two features were computed. The following approach could potentially yield more desirable results: modify the equations (3-3) and Eq. (3-4) by setting t to the day of the year corresponding to the timestamp and T to the number of days in a year. These modifications result in more samples with similar values, which can enhance the learning performance of the model on the yearly seasonality.



(a) Mean SHAP values **(b)** SHAP values

Figure 4-1: Feature importance of the XGBoost model expressed in SHAP values.

4-2 Hyperparameter optimisation results

As was mentioned in the previous chapter, the proposed deep learning model architectures were found through a hyperparameter optimisation algorithm. This section below summarises the random search results and the final parameter settings for both the multilayer perceptron (MLP) and long-short-term memory (LSTM) model.

4-2-1 Multilayer perceptron

Table 4-2 and Table 4-3 summarise the random search results obtained for the univariate and multivariate MLP model, respectively. Note that the mean squared error (MSE) loss is standardized.

Table 4-2: The hyperparameter optimisation results for the MLP model with univariate input data.

Data	Model	μ_{loss}	σ^2_{loss}	$\mu_{\text{val loss}}$	$\sigma^2_{\text{val loss}}$	Best validation loss	Best parameters
Univariate	1	0.0239	0.0000	0.0231	0.0000	0.0224	units_1: 25, learning_rate: 0.01
	2	0.0760	0.0199	0.0831	0.0146	0.0226	units_1: 5, dropout: 0.0, units_2: 75, learning_rate: 0.01
	3	0.0431	0.0187	0.0419	0.0174	0.0225	units_1: 5, num_additional_layers: 2, units_2: 85, units_3: 55, learning_rate: 0.01

Table 4-3: The hyperparameter optimisation results for the MLP model with multivariate input data.

Data	Model	μ_{loss}	σ^2_{loss}	$\mu_{\text{val loss}}$	$\sigma^2_{\text{val loss}}$	Best validation loss	Best parameters
Multivariate	1	0.0120	0.0000	0.0131	0.0000	0.0120	units_1: 10, learning_rate: 0.001
	2	0.0863	0.0293	0.0825	0.0266	0.0122	units_1: 70, dropout: 0.0, units_2: 95, learning_rate: 0.001
	3	0.0517	0.0375	0.0518	0.0345	0.0122	units_1: 30, num_additional_layers: 1, units_2: 35, units_3: 45, learning_rate: 0.001

No significant difference in the best validation score was found between the different model configurations. However, for the average and variance values over all trials, a remarkable difference between model 1 versus models 2 and 3 is observed. After manual exploration

of the different trials, it became apparent that the difference can primarily be ascribed to the search range of the second and third dense layer. Because when a trial initialises a zero neuron dense layer, no information is passed to the output layer, which causes a faulty trial. For model 2 specifically, this might also be due to high dropout rates within the search range; this is supported by the fact that the best validation loss is found at a zero dropout rate. A note of caution is due here since the (validation) loss between Table 4-2 and Table 4-3 cannot be directly compared. The underlying reason is that the univariate model has only one output, while the multivariate model has four. Therefore, the results on the power forecast were compared separately, which supported the expectation that the multivariate model outperforms the univariate model on the one-step-ahead power forecast.

Even though the best validation loss is slightly lower for model architecture 1, it was decided to continue with model 3 because of the expectation that this model will better map the complex task due to the additional layer and more significant number of neurons. The final architecture used to create the ensemble models consists of two dense layers with 30 and 35 neurons and has a learning rate of 0.001. The third layer that consists of 45 neurons is excluded from the model because the number of additional layers equals one for this configuration. This model architecture is from here onwards referred to as the $MLP - WPF$ model.

4-2-2 Long short term memory

Table 4-4 summarises the random search results obtained for the multivariate LSTM model.

Table 4-4: The hyperparameter optimisation results for the LSTM model.

Data	Model	μ_{loss}	σ^2_{loss}	$\mu_{\text{val loss}}$	$\sigma^2_{\text{val loss}}$	Best validation loss	Best parameters
Multivariate	1	0.0102	0.0000	0.0120	0.0000	0.0114	units_1: 24, learning_rate: 0.01
	2	0.0102	0.0000	0.0120	0.0000	0.0114	units_1: 16, units_2: 40, learning_rate: 0.001

Similar to the results of the multivariate MLP models, there is no significant difference observed related to model architecture. One unanticipated finding was that the total number of set trials was not reached for model 1 because the Oracle triggered an exit at trial 23. A possible explanation is that the random search algorithm could not generate a new set of hyperparameters from the search space. After exceeding a specific number of attempts, the Oracle stops. Nevertheless, this early stop of the random search does not seem to impact the results negatively.

The final architecture that was used to create the ensemble models consists of two LSTM layers with 16 and 40 neurons and has a learning rate of 0.001. This model architecture is from here onwards referred to as the $LSTM - WPF$ model.

4-3 Forecast quality

After the hyperparameter optimisation phase, the specified model architectures in section 4-2 were trained ten times on the train and validation set. The obtained ensemble models were used to construct four final models as described in section 3-3. These four models were evaluated on the test set using the quality performance metrics RMSE, mean absolute error (MAE) and mean bias error (MBE). The quality performance results are divided into three subsections. The first subsection presents the average results for specific forecast horizons. Subsequently, the focus is on the detailed results for every programme time unit (PTU) within the ultra-short-term wind power forecasting (UST-WPF) horizon. The final section examines the mean forecast bias.

4-3-1 Overall performance

Table 4-5 shows the average summary statistics on different forecast horizons for the benchmark and proposed models. The best score for every forecast horizon and performance metric is printed in bold. There are two important findings within these results. The most striking result to emerge from the data is that the $Eneco - LSTM_{p50}$ outperforms all the other models on the different forecast horizons, which is counterintuitive given that the MLP_{p50} model outperforms the $LSTM_{p50}$ model on the XBID and UST-WPF horizon. Illustrating an insignificant difference in performance between the two deep learning approaches, which confirms the findings of Liu et al. [29], but is in conflict with the findings of Li et al. [26]. The average reduction in RMSE between $Eneco - LSTM_{p50}$ and *Persistence* model on the forecast horizons from short to long are 15%, 22% and 31%, respectively. Compared to the *Eneco* model, the reductions are 27%, 16% and 4%, respectively. The second finding is that all other models, including the *Persistence* benchmark model, on average outperform the *Eneco* model until the XBID horizon on the MAE metric.

These results are likely to be related to the importance of real-time data for relatively short forecast horizons. This is supported by the decreasing RMSE reduction compared to the *Eneco* model. Moreover, also the *Persistence* model outperforms the *Eneco* on the shorter forecast horizons. It is essential to bear in mind that these results are the average performance metrics for specific horizons. Let us now consider the more detailed results of the different models.

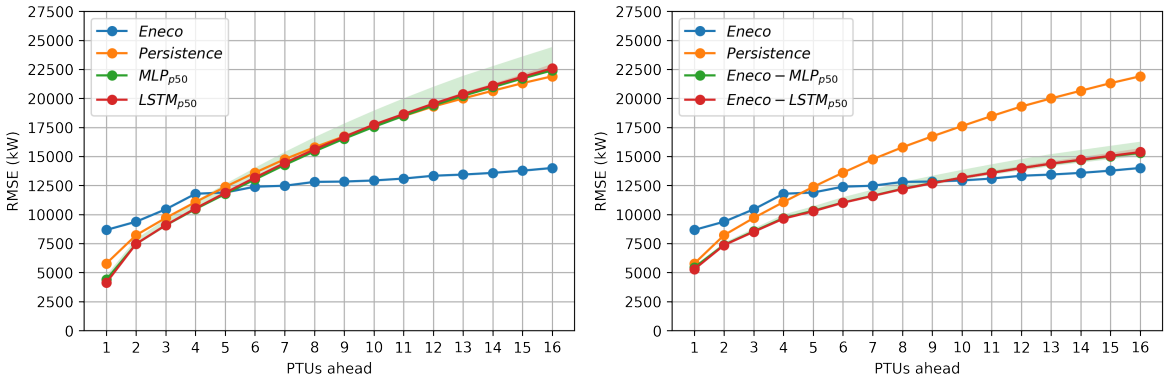
Table 4-5: Comparison of the average forecast accuracy of the benchmark models and proposed models on the test set and specified forecast horizons.

	National		**XBID**		**UST-WPF**	
Model	**RMSE (kW)**	**MAE (kW)**	**RMSE (kW)**	**MAE (kW)**	**RMSE (kW)**	**MAE (kW)**
Persistence	9433	5276	12006	6860	15457	9065
Eneco	10437	6394	11412	7098	12305	7794
MLP_{p50}	8634	5070	11382	6749	15200	9132
$LSTM_{p50}$	8618	5039	11446	6810	15306	9337
$Eneco - MLP_{p50}$	8279	4991	9885	6031	11820	7361
$Eneco - LSTM_{p50}$	**8218**	**4885**	**9843**	**5925**	**11804**	**7242**

4-3-2 Performance over forecast horizon

This subsection focuses on the quality performance of the models per lead time. The intention is to visualise when the forecast performance is on par with the benchmark models and illustrate the uncertainty related to the stochastic nature of the proposed deep learning models. Only the RMSE is reported to avoid approximately redundant results.

Figure 4-2a compares the performance of the two proposed deep learning models with the two benchmark models. The shaded areas indicate the 80% confidence interval based on the 10th and 90th percentile predictions of the ensemble models. Generally, the 10th and 90th percentile predictions are worse than the 50th percentile prediction; therefore, the shaded area is most prominent on the up side and barely visible on the bottom side. It can be observed that the stochastic element has more influence on the $MLP-WPF$ model than on the $LSTM-WPF$ model, which signifies that the $LSTM-WPF$ model is more stable than the $MLP-WPF$ model. Eventually, the performance of both models is slightly worse compared to the *Persistence* model, which means that the models do not have skill from 10 PTU onwards. Figure 4-2b compares the performance of the multivariate combination models with the two benchmark models. The shaded areas are based on the multivariate combination of the 10th and 90th percentile predictions with the *Eneco* model. The multivariate combination models outperform the *Persistence* model on the complete UST-WPF horizon and outperform the *Eneco* model up to the nine PTU horizon.

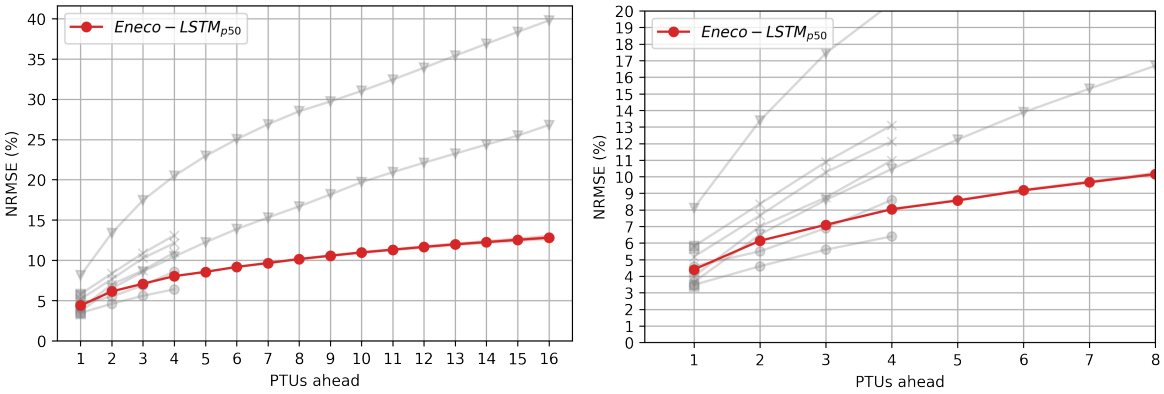


(a) MLP_{p50} and $LSTM_{p50}$ **(b)** $Eneco-MLP_{p50}$ and $Eneco-LSTM_{p50}$

Figure 4-2: The four proposed models compared to the *Persistence* and *Eneco* benchmark models.

Performance within the literature framework

The performance of the $Eneco-LSTM_{p50}$ can be compared with some of the models found in the literature through normalising the RMSE with the nominal farm capacity (i.e., normalized root-mean-square error (NRMSE)). Figure 4-3a and Figure 4-3b show the comparison for a 16 PTU and 8 PTU horizon, respectively. These plots must be interpreted with caution because these studies use different data sets. Moreover, the variation in sample size can strongly influence the results. Nevertheless, these figures still put the best performing model of this study into context.



(a) 16 PTU horizon **(b)** 8 PTU horizon

Figure 4-3: The $Eneco - LSTM_{p50}$ model compared to the results from the reviewed literature in section 2-3.

4-3-3 Forecast bias

The forecast bias indicates whether the model tends to overestimate or underestimate wind power production. Table 4-6 presents the MBE for the two benchmark models and four proposed models, where the value closest to zero is printed in bold per time interval. What stands out is that all models generally overforecast except for the $LSTM_{p50}$ model, which slightly underestimates the power production. From a forecaster's perspective the *Persistence* and $LSTM_{p50}$ perform best on the MBE metric.

Table 4-6: The mean bias error with an hourly frequency on the UST-WPF horizon expressed in kW.

Model	0-4 PTU	4-8 PTU	8-12 PTU	12-16 PTU
Persistence	69	**76**	**71**	57
Eneco	1411	2198	2869	3458
MLP_{p50}	461	694	983	1275
$LSTM_{p50}$	**-8**	-281	-294	**-4**
$Eneco - MLP_{p50}$	936	1446	1926	2367
$Eneco - LSTM_{p50}$	701	958	1287	1727

4-4 Forecast value

This section covers the results from the valuation model proposed in section 3-4. First, the overall cumulative value creation results are presented. Subsequently, the underlying driving factors for value creation are explored for three specific models.

4-4-1 Overall performance

Table 4-7 shows the cumulative forecast value relative to the *Eneco* model for the test set. The different forecast horizons have to be interpreted as follows: the five PTU forecast horizon leads to only one potential intraday trade per delivery hour. In comparison, the nine PTU forecast horizon leads to five potential intraday trades per delivery hour. These findings cannot be extrapolated into the future and might differ in practice. Nevertheless, four interesting findings result from this table. Firstly, all models produce a positive revenue. Secondly, the multivariate combination models have a significantly lower value creation, which is explainable through considering that the forecast is for 50% determined by the *Eneco* model. Thirdly, the increase in the forecast horizon improves the valuation results except for the final increment in the forecast horizon of the $LSTM_{p50}$ model. The increase in market liquidity may partly explain the relationship between forecast value and forecast horizon. Finally, these results further support the idea that a high forecast quality does not necessarily deliver the most value.

The cumulative forecast value results of the 10th or 90th percentile models are included in the appendix, see Table D-1. These results do not indicate a consistently better value creation for either the 10th or 90th percentile forecast. Therefore, based on these results, there is no clear relationship between forecast value and over or under forecasting.

The next section aims to provide more insight into what causes the cumulative value creation of the $Persistence$, MLP_{p50} and $Eneco - LSTM_{p50}$ models. Because the $Persistence$ model functions as a benchmark. The MLP_{p50} model is the best performing model on the European Cross-Border Intraday (XBID) horizon. Lastly, the $Eneco - LSTM_{p50}$ model is the most accurate model on the XBID horizon.

Table 4-7: Cumulative forecast value relative to the *Eneco* model expressed in euros over the test set on all forecast horizons between National and XBID.

Model	5 PTU	6 PTU	7 PTU	8 PTU	9 PTU
$Persistence$	35450	**52391**	**52771**	57901	67641
$Eneco$	0	0	0	0	0
MLP_{p50}	**37743**	51616	51781	**59410**	**69353**
$LSTM_{p50}$	33290	40028	41700	46356	46208
$Eneco - MLP_{p50}$	25664	35426	37458	41575	47750
$Eneco - LSTM_{p50}$	21608	27179	29027	32142	34867

4-4-2 Detailed valuation results

The first subsection displays the cumulative value creation of the $Persistence$, MLP_{p50} and $Eneco - LSTM_{p50}$ in more detail. On account of these results, the latter subsection focuses on one specific week in the test set.

The value creation for three different models

Figure 4-4 presents the cumulative value creation over the test set of the three selected models. Overall the models seem to generate a relatively small positive forecast value compared to

the *Eneco* model, but between 21/12/2020 and 27/12/2020, the value creation spikes.

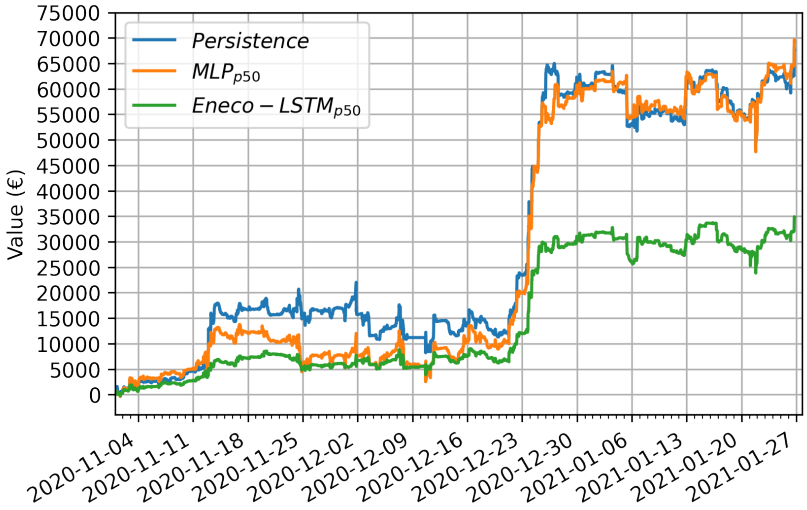
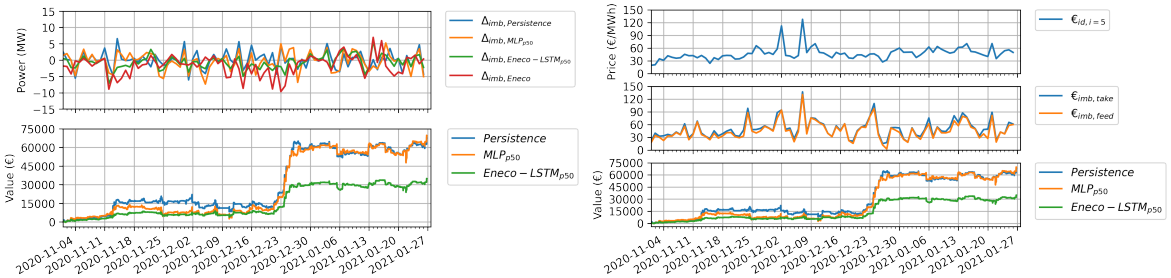


Figure 4-4: Cumulative value creation of the $Persistence$, MLP_{p50} and $Eneco - LSTM_{p50}$ model over the test set.

Some of the driving factors for this value creation are the imbalance volumes, the XBID horizon volume weighted average price (VWAP) and the imbalance prices. The daily mean of the imbalance volumes and prices were plotted on top of Figure 4-4 in Figure 4-5a and Figure 4-5b, respectively. The reason behind taking the daily mean is to make the data visually more appealing and interpretable. Based on both figures, it appears that most value creation is characterised by a significant short position of the *Eneco* model during relatively high imbalance prices compared to the intraday VWAP.

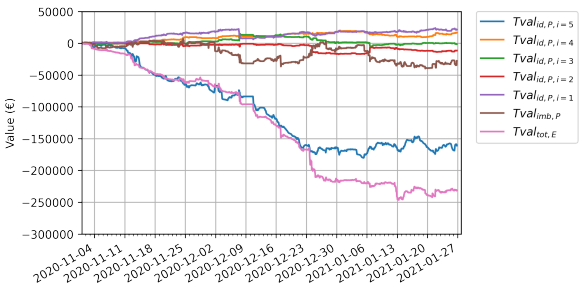


(a) Daily mean imbalance volume and cumulative value **(b)** Daily mean price data and cumulative value

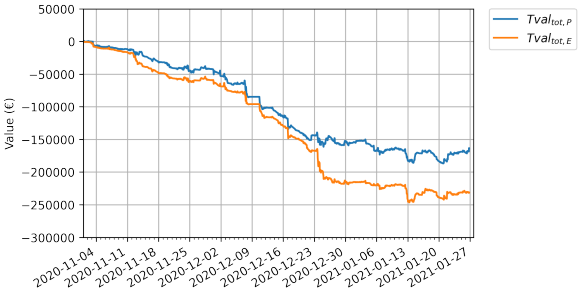
Figure 4-5: Cumulative value with the daily mean imbalance volumes and prices.

To further investigate whether this significant value creation can largely be ascribed to the short position of the *Eneco* model all trade value (TVAL) components are plotted separately in a cumulative fashion in Figure 4-6a, Figure 4-6c and Figure 4-6e. Furthermore, the cumulative $Tval_{tot,P}$ and $Tval_{tot,E}$ are plotted for the three models in Figure 4-6b, Figure 4-6d and Figure 4-6f.

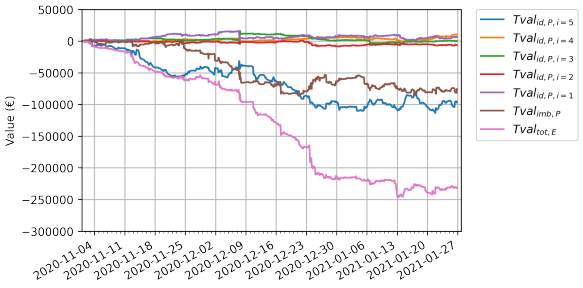
The figures show that the three examined models neither have significant intraday or imbalance costs during this specific week. Of these three models, the $Persistence$ model has the most negative intraday TVAL on the XBID horizon but has almost negligible costs related to imbalance. For the other two models, these parameters are of comparable magnitude, where the MLP_{p50} has a slightly lower imbalance cost compared to the intraday TVAL on the XBID horizon and the $Eneco - LSTM_{p50}$ vice versa. The imbalance costs of the $Eneco$ model seem to be significant, supporting the earlier findings, which pointed towards the short position of the model during this specific week. Following the findings in this section, the next section focuses on the $Tval_{tot,E}$ within this particular period at a higher granularity.



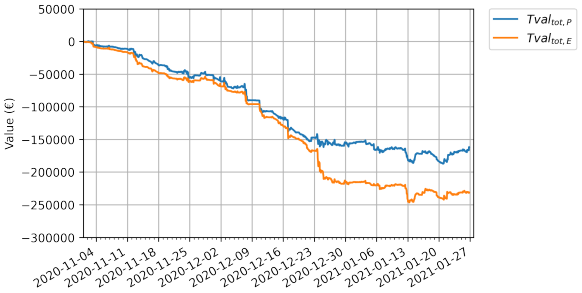
(a) Seperate TVAL components of the $Persistence$ and $Eneco$ model



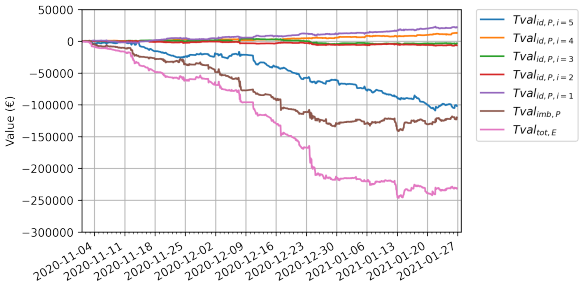
(b) TVAL of the $Persistence$ and $Eneco$ model



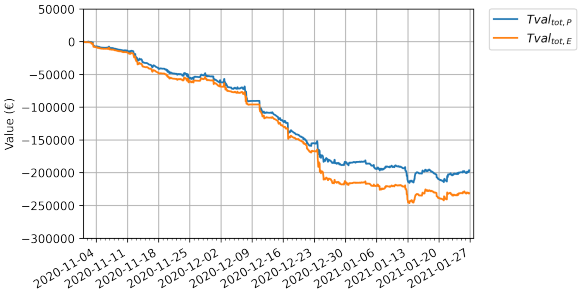
(c) Seperate TVAL components of the MLP_{p50} and $Eneco$ model



(d) TVAL of the MLP_{p50} and $Eneco$ model



(e) Seperate TVAL components of the $Eneco - LSTM_{p50}$ and $Eneco$ model



(f) TVAL of the $Eneco - LSTM_{p50}$ and $Eneco$ model

Figure 4-6: Detailed plots to visualise the separate TVAL components of the $Persistence$, MLP_{p50} and $Eneco - LSTM_{p50}$ model.

The value creation from 21/12/2020 until 27/12/2020

Based on the previous subsection, it was concluded that the significant change in value is mainly due to the high imbalance costs of the *Eneco* model. In Figure 4-7 the wind speed is visualised in the top plot, and in the bottom plot, the mean *Eneco* forecast in the last tradable PTU and the allocation data are shown. Three particular moments have been highlighted, during which a considerable difference between the *Eneco* forecast and the actual production is observed. The observed difference is most likely due to the sudden decrease in wind speed, while the wind speed is close to the rated output speed.

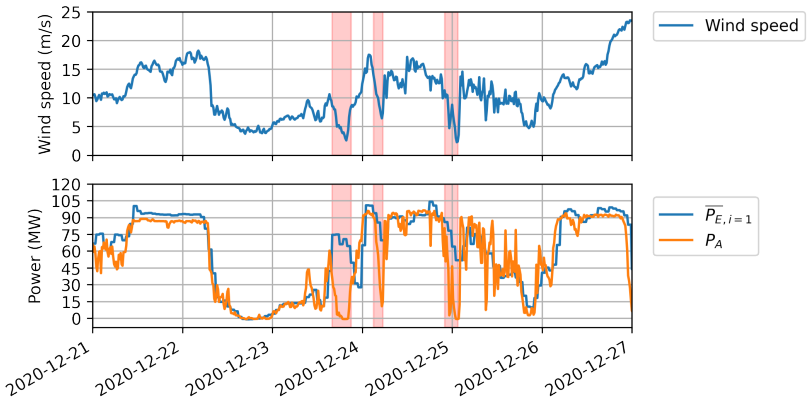


Figure 4-7: The mean *Eneco* forecast on the last tradable time and the allocation in the context of the measured wind speed from 21/12/2020 until 27/12/2020.

The delta between the *Eneco* forecast and the allocation is shown in the top plot of Figure 4-8. Combined with the respective imbalance price, this computes the $Tval_{tot,E}$, see the bottom plot of this figure. This combination of findings provides support for the importance of real-time data for the forecast value on the XBID horizon.

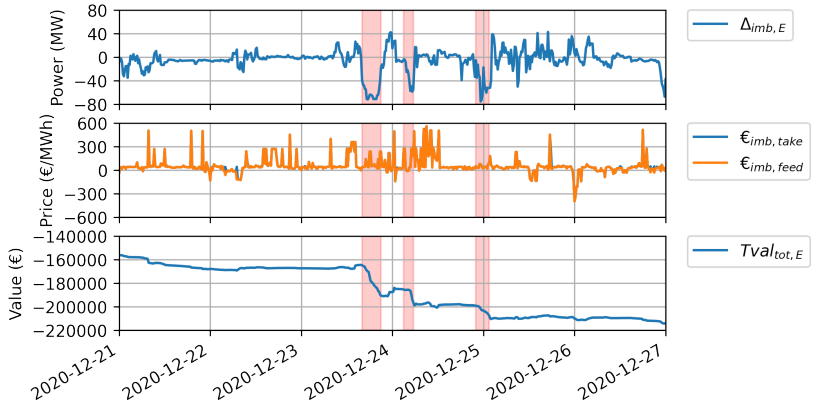


Figure 4-8: The imbalance volume of the *Eneco* model, imbalance prices and $Tval_{tot,E}$ accumulation that is a product of these volumes and prices from 21/12/2020 until 27/12/2020.

4-5 Case studies

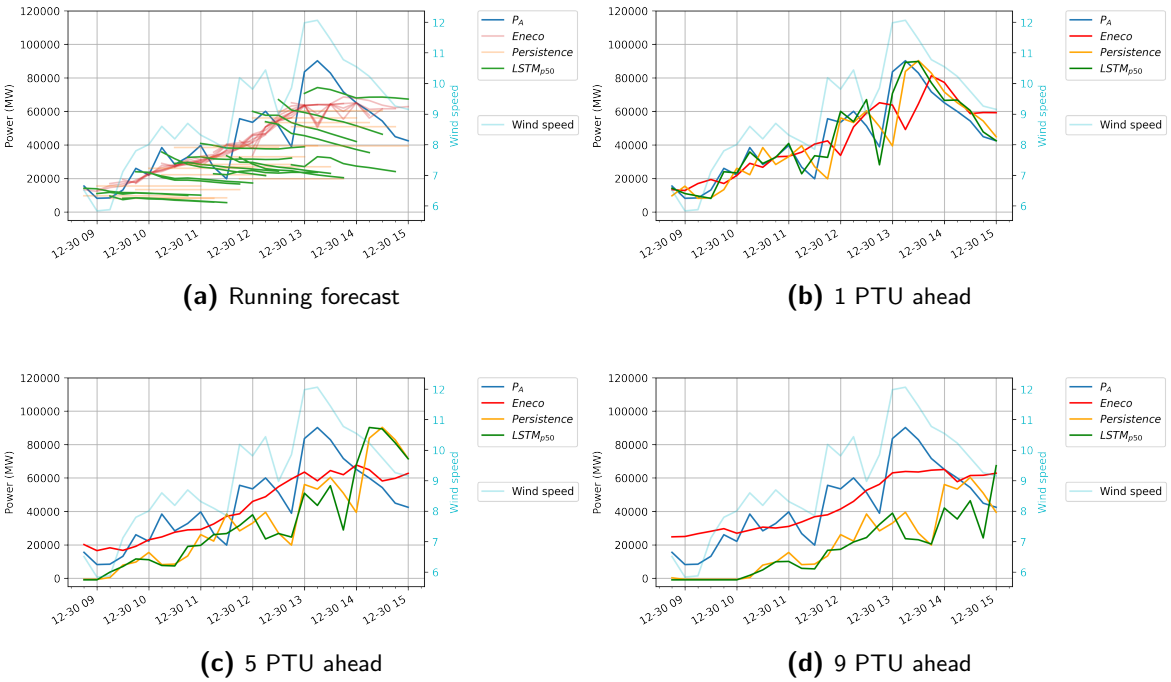
This final section aims to create insight into the forecast under specific conditions. All plots contain the allocation data ($\mathbf{P_A}$), measured wind speed and the two benchmark models. Because plotting all proposed models would be overwhelming, it was decided only to show the $LSTM_{p50}$ model on the nine PTU ahead horizon. As the multivariate combination between this and the *Eneco* model generates the best performing model, which is the $Eneco{-}LSTM_{p50}$ model. Plotting both separately creates insight into why the combination is beneficial for the forecast quality. Every figure contains four plots where plot **(a)** shows the running forecast with a 9 PTU ahead horizon and the plots **(b)**, **(c)** and **(d)** show the 1, 5, and 9 PTU ahead forecasts, respectively.

Firstly, two ramp-up and ramp-down events are shown. Subsequently, a consistently high and low wind speed case is considered. The definitions of ramp events in the papers of Cutler et al. [9], and Bossavy et al. [3] have functioned as an inspiration. The following guidelines were used to chose the four different ramp occurrences: a *quick ramp event* has approximately a 50% change in rated power within one hour while the *gradual ramp event* has approximately a 75% change in rated power within three hours. The consistent wind speed cases were found empirically.

4-5-1 Case 1: Ramp-up events

Case 1A: Ramp-up event within one hour

Its inconsistent increase characterises the ramp event on 30/12/2020. Figure 4-9 shows that in this specific case, the $LSTM_{p50}$ performs well on the first PTU ahead forecast. However, the model under-forecasts and does not recognise the up-going trend for multiple steps ahead. Because of this, the model performs similarly to the *Persistence* model.



(a) Running forecast

(b) 1 PTU ahead

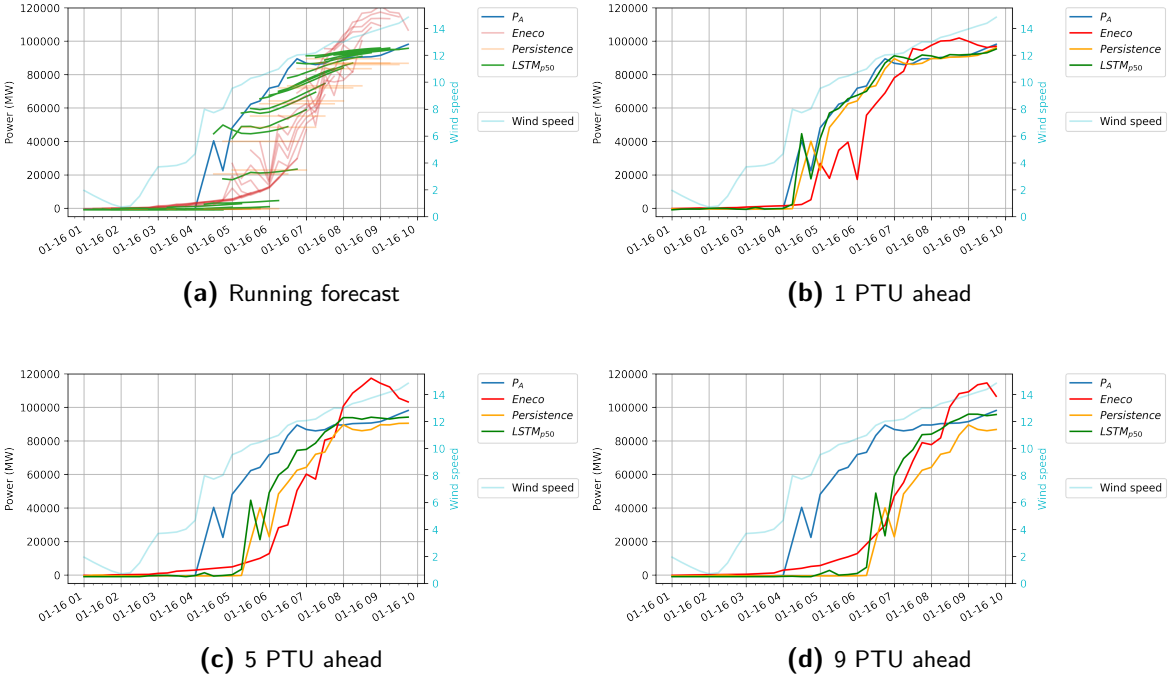
(c) 5 PTU ahead

(d) 9 PTU ahead

Figure 4-9: Ramp-up from 38 MW to 90 MW between 12:45 and 13:15 on 30/12/2020.

Case 1B: Ramp-up event within three hours

In contrast to earlier findings in the one-hour ramp-up case, the $LSTM_{p50}$ picks up the upward-trend within the three-hour ramp-up case, see Figure 4-10a. Therefore, it generally outperforms the two benchmark models. Moreover, the $LSTM_{p50}$ does not significantly overshoot, which is the case for the *Eneco* model. In this case, the value of high dependence on real-time data for short forecast horizons becomes apparent. A possible explanation for this might be the more consistent and gradual increase. Furthermore, the real-time wind speed ramps up in advance of the allocation data, which provides the $LSTM_{p50}$ model with ramp-up information.



(a) Running forecast **(b)** 1 PTU ahead

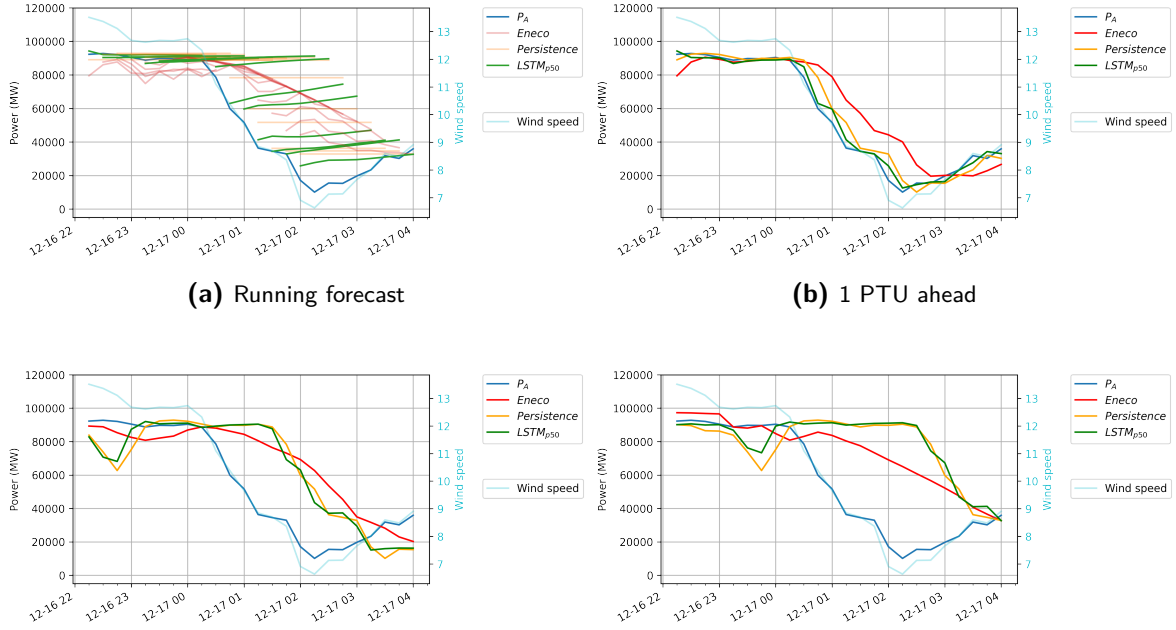
(c) 5 PTU ahead **(d)** 9 PTU ahead

Figure 4-10: Ramp-up from -0.1 MW to 90 MW between 04:00 and 07:00 on 16/01/2021.

4-5-2 Case 2: Ramp-down events

Case 2A: Ramp-down event within one hour

There are similarities between the $LSTM_{p50}$ behaviour in the ramp-up case 1A and the ramp-down case within one hour. Looking at the running forecast in Figure 4-11a the $LSTM_{p50}$ model only picks up the downward trend on the first PTU ahead forecast as it forecasts lower than the persistence model, which is more clearly visualised in Figure 4-11b. However, the forecast even slightly increases for the following PTUs ahead. Even though the increase the model still performs slightly better than persistence, but for the 5th and 9th PTU ahead, the results look somewhat similar to the persistence model. These results may support the influence of the wind speed feature on the forecast. In ramp-up case 1B, the wind speed preceded the allocation power, which informs the model on the ramp-up; this is not the case when the wind speed and allocation align. The inconsistent phase shift between the wind speed and allocation data is most likely due to the earlier mentioned issues related to the Breeze asset management system.



(a) Running forecast

(b) 1 PTU ahead

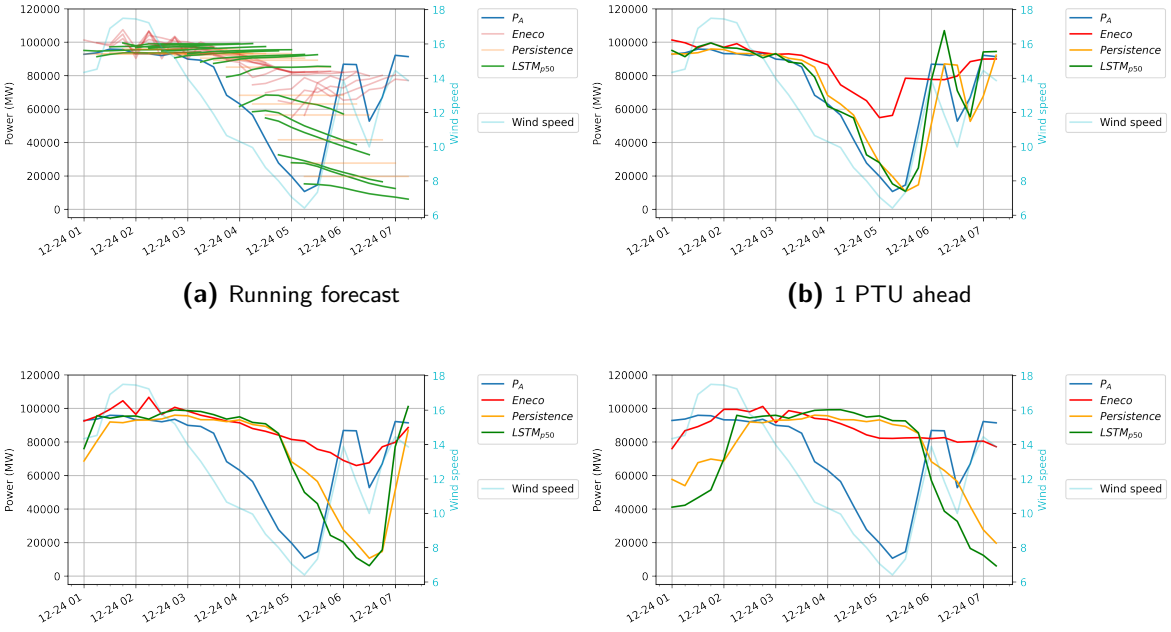
(c) 5 PTU ahead

(d) 9 PTU ahead

Figure 4-11: Ramp-down from 90 MW to 36 MW between 00:15 and 01:15 on 17/12/2020.

Case 2B: Ramp-down event within three hours

The combination of the gradual ramp-down and the preceding decreasing wind speed likely cause the $LSTM_{p50}$ to forecast lower than the *Persistence* model, which supports the earlier findings that the model has skill over *Persistence*, see Figure 4-12a. In Figure 4-12b it becomes clear that the *Eneco* model is already slightly modified with real-time data through the UST correction model. However, a more substantial correction would be beneficial. This perfectly illustrates the reason behind the improved performance of the $Eneco - LSTM_{p50}$ model.



(a) Running forecast

(b) 1 PTU ahead

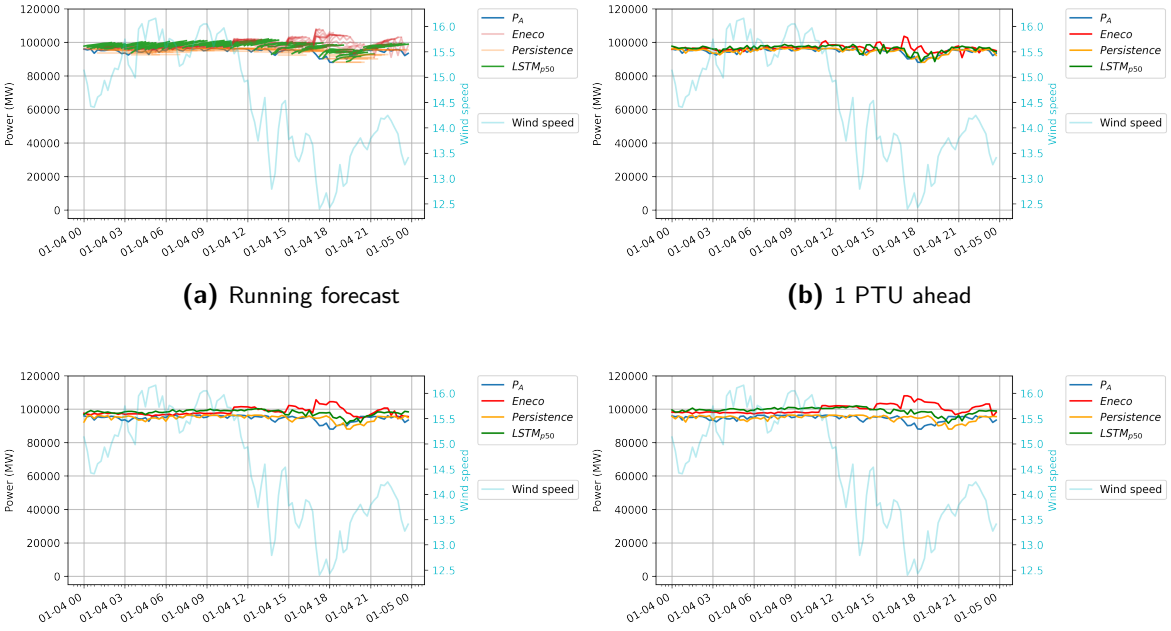
(c) 5 PTU ahead

(d) 9 PTU ahead

Figure 4-12: Ramp-down from 94 MW to 10 MW between 02:45 and 05:45 on 24/12/2020.

4-5-3 Case 3: Consistent high wind speeds

The difference between all models is relatively low during the consistently high wind speeds scenario. The most noticeable difference is the over-forecasting of the *Eneco* model around 18:00; this is most likely related to the numerical weather prediction (NWP) data used as an input for the *Eneco* model.



(a) Running forecast

(b) 1 PTU ahead

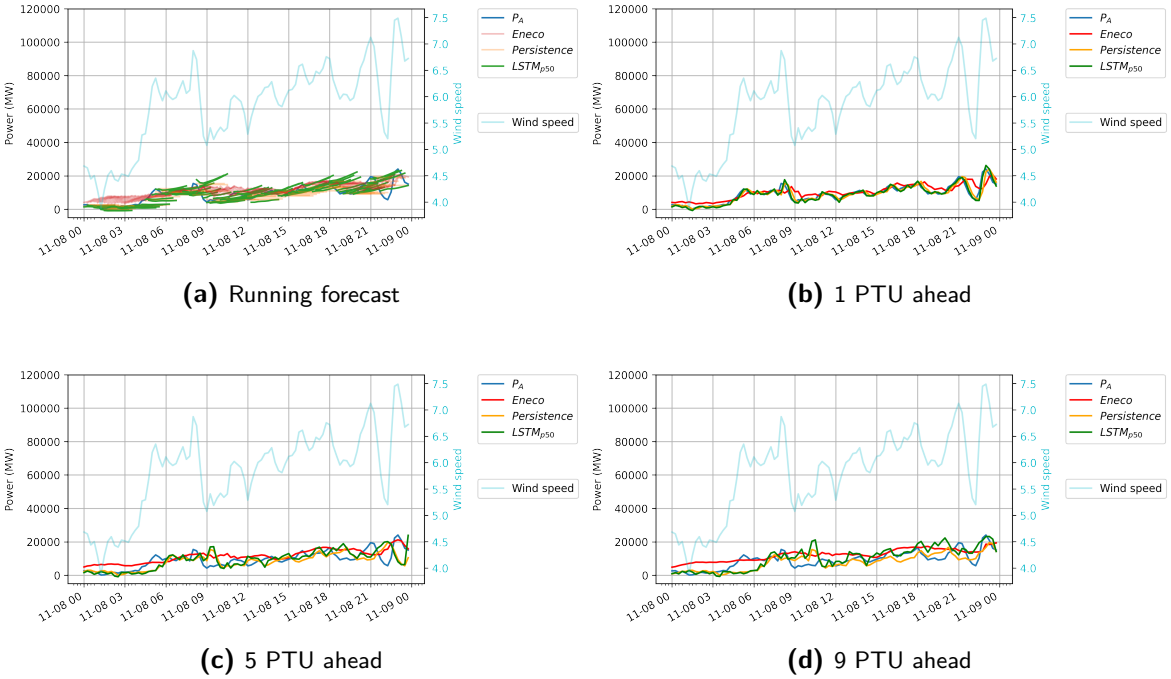
(c) 5 PTU ahead

(d) 9 PTU ahead

Figure 4-13: Consistent high wind speeds on 04/01/2021.

4-5-4 Case 4: Consistent low wind speeds

The consistent low wind speed case has a slightly increasing trend, which the $LSTM_{p50}$ model seems to forecast, see Figure 4-14a. Overall the models perform very similarly on this section of the test set.



(a) Running forecast

(b) 1 PTU ahead

(c) 5 PTU ahead

(d) 9 PTU ahead

Figure 4-14: Consistent low wind speeds on 08/11/2020.

Chapter 5

Conclusion & Recommendations

5-1 Conclusions

The ongoing large scale adoption of wind power increases the associated risks related to the variability. An essential way to mitigate these risks is to forecast production accurately. This has been the motivation for this research into wind power forecasting (WPF), specifically ultra-short-term wind power forecasting (UST-WPF), due to its commercial and technical relevance.

Two open areas of research have influenced the direction of this research. Firstly, the desire for a more practical application of forecast methods considering both accuracy and value. Most state-of-the-art research is conducted from a forecaster's perspective, focusing on reducing the standard error metrics. However, from a forecast user perspective, it is evenly important to generate value with a forecast model. Secondly, the increasing amount of available data and the developments within the domain of artificial intelligence (AI) over the past decades present opportunities for forecasting. Until now, deep learning has not yet delivered exceptional results in forecasting compared to other research fields. Therefore, the research goal of this project has been to explore the potential of deep learning models to increase both forecast quality and value.

The approach has been the development of four recursive UST-WPF models for Princess Amalia Wind Farm (PAWP) with a 16 programme time unit (PTU) forecast horizon and a forecast frequency of 1 PTU. Model 1 and model 2 use a multilayer perceptron (MLP) and a long-short-term memory (LSTM) architecture, respectively. Both models only use real-time data to forecast wind power. After finding the optimal hyperparameters through a random search algorithm, these models were trained ten times to compute the 10th percentile, median and 90th percentile forecast. This makes the models more robust and quantifies the stochastic nature of deep learning models. The other two models are a multivariate combination of the median ensemble forecast models with the currently operational ultra-short-term (UST) corrected numerical weather prediction (NWP) based model (i.e., the *Eneco* model). The four models that result from this approach are the MLP_{p50}, $LSTM_{p50}$, $Eneco-MLP_{p50}$ and

$Eneco - LSTM_{p50}$, which were compared to two benchmark models: a *Persistence* and the *Eneco* model. Additionally, a novel framework was designed to evaluate the forecast value relative to the *Eneco* model on various forecast horizons.

Based on the obtained results, the following five conclusions can be drawn. Firstly, it appears that both proposed deep learning methods, MLP_{p50} and $LSTM_{p50}$, outperform the *Persistence* benchmark model on a nine PTU ahead horizon. Secondly, the multivariate combination of these methods with the operational *Eneco* model, $Eneco - MLP_{p50}$ and $Eneco - LSTM_{p50}$, outperform the *Eneco* model on a nine PTU ahead horizon. Thirdly, the difference in performance between the MLP and LSTM is remarkably small in the proposed configurations. However, the LSTM model does show to be more consistent compared to the MLP model. Fourthly, all proposed models have a smaller bias than the *Eneco* model, which is a desirable model characteristic. Finally, all proposed models generate positive value relative to the *Eneco* model, but the statistically best model does not necessarily generate the most value. To summarise, the results indicate that the proposed deep learning models can contribute both in quality and value up to 9 PTUs ahead.

5-2 Drawbacks and limitations

During the project, the following drawbacks and limitations were encountered:

- **Length data set.** The performance of deep learning methods is strongly related to the amount of training data. The number of training samples is constrained by the implementation date of the Breeze asset management system. Because the meteorological observation features were only available from this data source at Eneco, consequently, not the entire length of historical power measurements could be used.

- **Data quality.** The data in Breeze is most likely inconsistently lagged. Considering that accurate lagged real-time power measurements are an essential feature in UST-WPF the study has been limited to regel- en reservevermogen (RRV) wind farms that do have an accurate real-time power datastream.

- **Lack of generalisation.** This methodology has only been applied to one particular offshore wind farm. Therefore it is unknown how these conclusions generalise to other RRV wind farms. Furthermore, the short duration of the test set related to the initialisation date of the *Eneco* model logging means that these results have not yet been validated for any spring or summer months.

- **Limited design iterations.** Training of deep learning models is a time-consuming process, even with high-performance computational resources. Not only the training but also generating the running forecast over the test set is time-intensive. The combination of both has limited the number of design iterations.

5-3 Recommendations

5-3-1 Business recommendations

The core challenge for a utility company related to this research is to bridge the divide between theoretical improvements in research and an operational workflow. Because of the following two reasons, there is some resistance to implement deep learning models. First, a utility company values reliability above all else. Therefore, the interpretability of the forecast will be crucial for adoption within this industry. More research into the explainability of deep learning models is recommended. Secondly, there is still, to some extent, a gap concerning innovation and integration structures. Developing a deep learning workflow and a standard parallel test environment for quality and value could aid the swift implementation. Regarding forecast value, the theoretical value model proposed in this thesis can function as a basis to build on. Nevertheless, the implications of the assumptions made to conduct the financial analysis need to be critically examined. A sensitivity analysis on these assumptions can be a future research topic.

For Eneco specifically, it is recommended to address the four limitations and drawbacks discussed in the previous subsection. In order to do this, the following opportunities are suggested:

- Investigate whether the additional historical power data weighs more heavily on model performance than the features from Breeze.
- Start the conversation with Greenbyte about data consistency. Maybe they can improve the software and replace the anomalous data.
- Schedule a new analysis when a full year of data is available to evaluate the model performance on all seasons. Furthermore, the developed models can be validated on similar wind farms within the portfolio.
- Improve the designed recursive forecasting framework; this reduces the evaluation time on the test set.

5-3-2 Academic recommendations

In reflection of developing the deep learning WPF models, the following avenues for future research are summarised as follows:

1. Considering that the multivariate combination models outperform all other models from two PTUs ahead onward. Incline that the inclusion of NWP data or other available forecasts to the input features might improve the model performance. Future research could, for instance, quantify the importance of NWP on the UST-WPF horizon.
2. This research takes into account the stochastic element of deep learning models and quantifies its influence on the UST-WPF horizon. Further research can include NWP ensemble data to provide an indication of meteorological uncertainty and subsequently compare the magnitude of the two sources of forecast uncertainty.

3. The performance of the proposed LSTM model is more stable, but it does not seem to outperform the MLP model significantly. A greater focus on the difference in results between various LSTM architectures could produce interesting findings. Some suggestions are implementing the recursive model from this thesis, an encoder-decoder model and a multi-step vector output model.

4. The selection process of the number of lagged inputs in this thesis were influenced by results found in literature and the results from the partial autocorrelation function (PACF) analysis. Further research is required to determine whether an increased number of lagged values improve the results of either the MLP or LSTM model.

5. A fundamental approach to capture the yearly seasonality has a negligible effect on the forecast. Further research could be conducted to determine the effectiveness of alternative methods to incorporate seasonality like differencing, time-series decomposition and one-hot encoding.

6. Several questions remain to be answered regarding the results of the random search hyperparameter optimisation of this study. A natural progression of this work is to analyse the results for more complex architectures, additional input features and more advanced optimisation algorithms, like Bayesian optimisation.

7. In the studied literature, the inclusion of features is often based on the correlation with wind power. A further study could assess feature importance in a more modular fashion to quantify the relationship between specific input features and forecast quality and value.

8. Based on the studied literature, the mean squared error (MSE) loss function was applied. Nevertheless, different loss functions might better approximate the actual cost function, which is most likely asymmetric. Future studies can investigate the influence of different loss functions on value creation by implementing the proposed value framework.

9. The multivariate combination of models is weighted equally. However, based on the obtained results, it seems interesting to pursue the development of an algorithm that optimizes the weights between both models for every PTU.

Appendix A

Initial selection Eneco wind portfolio

Table A-1: The Eneco wind portfolio filtered on the initial selection criteria.

Country	Subgroup	Site name	Nominal Power (MW)	Lat	Lon
NL	Onshore	Slufter	21.6	51.93196	4.014845
		Herkingen II	8.25	51.70857	4.11307
		Delfzijl ZO	15.13	53.28468	6.96886
		Waalwijk	7.5	51.70916	5.072338
		Acrres	2.3	52.51553	5.55035
		Romerswaal	17.5	51.41659	4.230881
		Hoevensche Beemden	15	51.60921	4.591458
		Fujifilm	10	51.59656	5.017764
		Houten	6	52.01714	5.144442
		Delfzijl-Noord	62.7	53.31766	6.983167
		Laarakkerdijk	10	51.33191	5.136647
		Sabinapolder	9	51.67193	4.398243
		de Kroeten	0.85	51.63206	4.708825
		Anna-Mariapolder	14.4	51.38326	4.265265
		Kloosterboer I	2.05	51.47829	3.70778
		Kloosterboer II	6.9	51.47167	3.69865
		Autena	9	51.96966	5.106693
		WP Boerderijweg	9.2	51.30341	5.939264
		WP Dalfsen	9.9	52.56118	6.213787
		WP IJslandweg	2	51.44052	3.736372
		WP Oesterdam	5	51.45413	4.232188
		WP Tolhuis	9.9	52.56297	6.20628
		WP Van Gogh	11.5	51.61489	4.609872
		WP Zuidwal I	15	51.9791	4.061
		WP Zuidwal II	9	51.9825	4.0442
		WP Landtong II	6	51.93861	4.18712
NL	Offshore	Prinses amaliawindpark	120	52.58755	4.224012
		Luchterduinen	129	52.40463	4.162962
		Borssele	365.75	51.67318	2.887471

Appendix B

Valuation model trade cycle

Start trade time	End trade time (National)	End trade time (XBID)	Available data	1 PTU ahead	2 PTU ahead	3 PTU ahead	4 PTU ahead	5 PTU ahead	6 PTU ahead	7 PTU ahead	8 PTU ahead	9 PTU ahead
00:46:00	00:55:00	01:00:00	Delivery time	01/11/2020 00:45	01/11/2020 01:00	01/11/2020 01:15	01/11/2020 01:30	01/11/2020 01:45	01/11/2020 02:00	01/11/2020 02:15	01/11/2020 02:30	01/11/2020 02:45
			01/11/2020 00:30	96040	93224	70124	49764	43536	37652	39358	61200	62924
01:01:00	01:15:00	x	Delivery time	01/11/2020 01:00	01/11/2020 01:15	01/11/2020 01:30	01/11/2020 01:45	01/11/2020 02:00	01/11/2020 02:15	01/11/2020 02:30	01/11/2020 02:45	01/11/2020 03:00
			01/11/2020 00:45	93224	70124	49764	43536	37652	39358	61200	62924	64160
01:16:00	01:30:00	x	Delivery time	01/11/2020 01:15	01/11/2020 01:30	01/11/2020 01:45	01/11/2020 02:00	01/11/2020 02:15	01/11/2020 02:30	01/11/2020 02:45	01/11/2020 03:00	01/11/2020 03:15
			01/11/2020 01:00	70124	49764	43536	37652	39358	61200	62924	64160	84676
01:31:00	01:45:00	x	Delivery time	01/11/2020 01:30	01/11/2020 01:45	01/11/2020 02:00	01/11/2020 02:15	01/11/2020 02:30	01/11/2020 02:45	01/11/2020 03:00	01/11/2020 03:15	01/11/2020 03:30
			01/11/2020 01:15	49764	43536	37652	39358	61200	62924	64160	84676	89100
01:46:00	01:55:00	02:00:00	Delivery time	01/11/2020 01:45	01/11/2020 02:00	01/11/2020 02:15	01/11/2020 02:30	01/11/2020 02:45	01/11/2020 03:00	01/11/2020 03:15	01/11/2020 03:30	01/11/2020 03:45
			01/11/2020 01:30	43536	37652	39358	61200	62924	64160	84676	89100	83440

Legend:
- No trading possible
- National
- XBID

Figure B-1: The trading cycle of the valuation model explained until the 9 programme time unit (PTU) ahead forecast.

Appendix C

Forecast quality

Table C-1: The forecast results on the test set expressed in root-mean-square error (RMSE) over the whole ultra-short-term wind power forecasting (UST-WPF) horizon.

PTU ahead	$LSTM_{p10}$	$LSTM_{p50}$	$LSTM_{p90}$	MLP_{p10}	MLP_{p50}	MLP_{p90}	$Eneco$	$Eneco-LSTM_{p10}$	$Eneco-LSTM_{p50}$	$Eneco-LSTM_{p90}$	$Eneco-MLP_{p10}$	$Eneco-MLP_{p50}$	$Eneco-MLP_{p90}$	$Persistence$
1	4124	4126	4151	4451	4393	4760	8672	5276	5285	5314	5379	5431	5570	5779
2	7444	7471	7495	7525	7462	7884	9381	7344	7364	7391	7316	7389	7575	8222
3	9067	9091	9128	9188	9072	9625	10451	8491	8515	8560	8478	8557	8802	9710
4	10510	10533	10565	10640	10475	11175	11787	9618	9646	9697	9614	9692	9998	11075
5	11845	11870	11905	11962	11770	12632	11896	10244	10280	10346	10234	10324	10711	12381
6	13153	13183	13222	13239	13022	14025	12396	10974	11017	11097	10929	11031	11493	13605
7	14405	14441	14485	14510	14272	15384	12478	11545	11599	11691	11499	11615	12144	14759
8	15557	15603	15657	15709	15440	16648	12810	12125	12190	12300	12087	12216	12796	15789
9	16640	16698	16764	16810	16529	17831	12838	12607	12686	12812	12552	12709	13336	16730
10	17674	17744	17821	17849	17549	18950	12931	13081	13173	13317	12994	13168	13842	17615
11	18568	18653	18748	18827	18496	20002	13097	13478	13586	13750	13415	13600	14325	18479
12	19438	19540	19653	19715	19390	20997	13338	13870	13996	14180	13789	14008	14781	19301
13	20256	20373	20508	20580	20225	21939	13440	14242	14385	14590	14156	14373	15201	19996
14	20978	21115	21272	21353	20968	22776	13586	14559	14718	14944	14468	14688	15557	20659
15	21692	21859	22083	22092	21717	23611	13776	14870	15051	15318	14761	15011	15916	21311
16	22358	22591	22979	22800	22416	24422	14008	15160	15379	15718	15045	15310	16268	21904

Appendix D

Cumulative forecast value

Table D-1: Cumulative forecast value for all models relative to the $Eneco$ model expressed in euros over the test set on all forecast horizons between National and European Cross-Border Intraday (XBID).

Model	**PTU 5**	**PTU 6**	**PTU 7**	**PTU 8**	**PTU 9**
$LSTM_{p10}$	31930	40750	43338	48564	47242
$LSTM_{p50}$	33290	40028	41700	46356	46208
$LSTM_{p90}$	35438	40452	41492	46494	48049
MLP_{p10}	27752	52834	56808	66382	66723
MLP_{p50}	37743	51616	51781	59410	69353
MLP_{p90}	48539	46459	43113	47842	70739
$Eneco$	0	0	0	0	0
$Eneco - LSTM_{p10}$	20511	27616	30009	33226	35854
$Eneco - LSTM_{p50}$	21608	27179	29027	32142	34867
$Eneco - LSTM_{p90}$	22859	27721	29330	32611	35684
$Eneco - MLP_{p10}$	19504	35114	38666	44212	46881
$Eneco - MLP_{p50}$	25664	35426	37458	41575	47750
$Eneco - MLP_{p90}$	30912	32345	31565	34227	44677
$Persistence$	35450	52391	52771	57901	67641

Bibliography

[1] Sanjeev Aggarwal. Wind Power Forecasting: A Review of Statistical Models. *International Journal of Energy Sciences*, 3:1–10, February 2013.

[2] R. J. Bessa, V. Miranda, A. Botterud, and J. Wang. 'Good' or 'bad' wind power forecasts: a relative concept. *Wind Energy*, 14(5):625–636, 2011. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/we.444.

[3] Arthur Bossavy, Robin Girard, and Georges Kariniotakis. Forecasting Uncertainty Related to Ramps of Wind Power Production. *EWEC 2010*, page 10, 2010.

[4] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, Greta M. Ljung, and Greta M Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, Incorporated, New York, UNITED STATES, 2015.

[5] Jason Brownlee. *Introduction to Time Series Forecasting with Python*. Jason Brownlee, 1.11 edition, 2020.

[6] J. P. S. Catalão, H. M. I. Pousinho, and V. M. F. Mendes. Short-term wind power forecasting in Portugal by neural networks and wavelet transform. *Renewable Energy*, 36(4):1245–1251, April 2011.

[7] Wen-Yeau Chang. A Literature Review of Wind Forecasting Methods. *Journal of Power and Energy Engineering*, 02(04):161–168, 2014.

[8] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, August 2016. Association for Computing Machinery.

[9] Nicholas Cutler, Merlinde Kay, Kieran Jacka, and Torben Skov Nielsen. Detecting, categorizing and forecasting large ramps in wind farm power output using meteorological observations and WPPT. *Wind Energy*, 10(5):453–470, 2007. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/we.235.

[10] Harsh S. Dhiman and Dipankar Deb. A Review of Wind Speed and Wind Power Forecasting Techniques. *arXiv:2009.02279 [cs, eess]*, September 2020. arXiv: 2009.02279.

[11] K. Dragomiretskiy and D. Zosso. Variational Mode Decomposition. *IEEE Transactions on Signal Processing*, 62(3):531–544, February 2014. Conference Name: IEEE Transactions on Signal Processing.

[12] Isaac Gendler. Fourier transform, August 2017.

[13] Gregor Giebel and Georges Kariniotakis. *Wind power forecasting - a review of the state of the art*. Elsevier - Woodhead Publishing, June 2017. Pages: Chapter 3.

[14] Li Han, Rongchang Zhang, Xuesong Wang, Achun Bao, and Huitian Jing. Multi-step wind power forecast based on VMD-LSTM. *IET Renewable Power Generation*, 13(10):1690–1700, 2019. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1049/iet-rpg.2018.5781.

[15] Shahram Hanifi, Xiaolei Liu, Zi Lin, and Saeid Lotfian. A Critical Review of Wind Power Forecasting Methods—Past, Present and Future. *Energies*, 13(15):3764, January 2020. Number: 15 Publisher: Multidisciplinary Digital Publishing Institute.

[16] S. E. Haupt, M. Garcia Casado, M. Davidson, J. Dobschinski, P. Du, M. Lange, T. Miller, C. Mohrlen, A. Motley, R. Pestana, and J. Zack. The Use of Probabilistic Forecasts: Applying Them in Theory and Practice. *IEEE Power and Energy Magazine*, 17(6):46–57, November 2019. Conference Name: IEEE Power and Energy Magazine.

[17] Sue Ellen Haupt, Pedro A. Jiménez, Jared A. Lee, and Branko Kosović. Principles of meteorology and numerical weather prediction. In *Renewable Energy Forecasting*, pages 3–28. Elsevier, 2017.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, Santiago, Chile, December 2015. IEEE.

[19] Norden E. Huang, Zheng Shen, Steven R. Long, Manli C. Wu, Hsing H. Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung, and Henry H. Liu. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1971):903–995, March 1998.

[20] Yuansheng Huang, Lei Yang, Shijian Liu, and Guangli Wang. Multi-Step Wind Speed Forecasting Based On Ensemble Empirical Mode Decomposition, Long Short Term Memory Network and Error Correction Strategy. *Energies*, 12:1822, May 2019.

[21] IEC. IEC TR 63043:2020. Technical report, IEC, November 2020.

[22] International Energy Agency. Renewables 2020 Analysis and forecast to 2025. *IEA*, page 172, 2020.

[23] Y. Ju, G. Sun, Q. Chen, M. Zhang, H. Zhu, and M. U. Rehman. A Model Combining Convolutional Neural Network and LightGBM Algorithm for Ultra-Short-Term Wind Power Forecasting. *IEEE Access*, 7:28309–28318, 2019. Conference Name: IEEE Access.

[24] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, January 2017. arXiv: 1412.6980.

[25] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[26] J. Li, D. Geng, P. Zhang, X. Meng, Z. Liang, and G. Fan. Ultra-Short Term Wind Power Forecasting Based on LSTM Neural Network. In *2019 IEEE 3rd International Electrical and Energy Conference (CIEEC)*, pages 1815–1818, September 2019.

[27] Bryan Lim, Sercan O. Arik, Nicolas Loeff, and Tomas Pfister. Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting. *arXiv:1912.09363 [cs, stat]*, September 2020. arXiv: 1912.09363.

[28] Rongsheng Liu, Minfang Peng, and Xianghui Xiao. Ultra-Short-Term Wind Power Prediction Based on Multivariate Phase Space Reconstruction and Multivariate Linear Regression. *Energies*, 11:2763, October 2018.

[29] Yao Liu, Lin Guan, Chen Hou, Hua Han, Zhangjie Liu, Yao Sun, and Minghui Zheng. Wind Power Short-Term Prediction Based on LSTM and Discrete Wavelet Transform. *Applied Sciences*, 9:1108, March 2019.

[30] S. Lu. Multi-Step Ahead Ultra-Short-Term Wind Power Forecasting Based on Time Series Analysis. In *2020 International Conference on Computer Information and Big Data Applications (CIBDA)*, pages 430–434, April 2020.

[31] Scott Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. *arXiv:1705.07874 [cs, stat]*, November 2017. arXiv: 1705.07874 version: 2.

[32] Markus Löning, Anthony Bagnall, Sajaysurya Ganesh, Viktor Kazakov, Jason Lines, and Franz J. Király. sktime: A Unified Interface for Machine Learning with Time Series. *arXiv:1909.07872 [cs, stat]*, September 2019. arXiv: 1909.07872.

[33] Jorge Maldonado-Correa, JC Solano, and Marco Rojas-Moncayo. Wind power forecasting: A systematic literature review. *Wind Engineering*, page 0309524X19891672, December 2019. Publisher: SAGE Publications.

[34] Jakob W. Messner, Pierre Pinson, Jethro Browell, Mathias B. Bjerregård, and Irene Schicker. Evaluation of wind power forecasts—An up-to-date view. *Wind Energy*, 23(6):1461–1481, 2020. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/we.2497.

[35] Eiman Mohammed, S. Wang, and J. Yu. Ultra-Short-Term Wind Power Prediction Using a Hybrid Model. *IOP Conference Series: Earth and Environmental Science*, 63:012005, May 2017.

[36] Corinna Mohrlen and Ricardo J Bessa. Understanding Uncertainty: the difficult move from a deterministic to a probabilistic world. *IEA Wind Task 36*, page 6, 2018.

[37] Aileen Nielsen. *Practical Time Series Analysis: Prediction with Statistics and Machine Learning*. O'Reilly Media, Sebastopol, CA, 1st edition edition, November 2019.

[38] Chris Olah. Understanding LSTM Networks, August 2015.

[39] Madasthu Santhosh, Chintham Venkaiah, and D. M. Vinod Kumar. Current advances and approaches in wind speed and wind power forecasting for improved renewable energy integration: A review. *Engineering Reports*, 2(6), June 2020. Publisher: John Wiley & Sons, Ltd.

[40] United Nations. Adoption of the Paris Agreement, December 2015.

[41] vikancha MSFT. NCv2-series - Azure Virtual Machines.

[42] Claire Vincent and Pierre-Julien Trombe. Forecasting intrahourly variability of wind generation. In *Renewable Energy Forecasting: From Models to Applications*, pages 219–233. Woodhead Publishing, December 2017. Journal Abbreviation: Renewable Energy Forecasting: From Models to Applications.

[43] Yusen Wang. Short-term Power Load Forecasting Based on Machine Learning. *KTH, School of Electrical Engineering and Computer Science (EECS)*, page 46, June 2020.

[44] Wind Europe. Wind energy in Europe 2020 Statistics and the outlook for 2021-2025, 2021.

[45] Steven H. Young and J. Zack. Impact of Targeted Measurements and Advanced Machine Learning Techniques on 0-3-h Ahead Rapid Update Wind Power and Ramp Rate Forecasts in the Tehachapi Wind Resource Area of California. In *Impact of Targeted Measurements and Advanced Machine Learning*. AMS, January 2018.

[46] Yinghui Zhang, Shiyuan Zhou, Ziqiang Zhang, Liang Yan, and Li Liu. Design of an Ultra-Short-Term Wind Power Forecasting Model Combined with CNN and LSTM Networks. In C. H. Wu, Srikanta Patnaik, Florin Popentiu Vlãdicescu, and Kazumi Nakamatsu, editors, *Recent Developments in Intelligent Computing, Communication and Devices*, volume 1185, pages 141–145. Springer Singapore, Singapore, 2021. Series Title: Advances in Intelligent Systems and Computing.

[47] Baobin Zhou, Che Liu, Jianjing Li, Bo Sun, and Jun Yang. A Hybrid Method for Ultrashort-Term Wind Power Prediction considering Meteorological Features and Seasonal Information, September 2020. ISSN: 1024-123X Pages: e1795486 Publisher: Hindawi Volume: 2020.

Glossary

List of Acronyms

ACF	autocorrelation function
ADF	Augmented Dickey-Fuller
AI	artificial intelligence
ANN	artificial neural network
API	Application Programming Interface
AR	AutoRegressive
ARIMA	autoregressive integrated moving average
BRP	Balancing Responsible Party
BRPs	Balancing Responsible Parties
CNN	convolutional neural network
DLVM	Deep Learning Virtual Machine
DNN	deep neural net
DSVM	Data Science Virtual Machine
EMD	empirical mode decomposition
EEMD	ensemble empirical mode decomposition
GPU	Graphical Processing Unit
GRU	gated recurrent unit
I	Integrated
IDE	Integrated Development Environment
IEC	International Electrotechnical Commission
IID	independent and identically distributed
IMFs	intrinsic mode functions
IT	information technology
MA	Moving Average

MAE mean absolute error
MAPE mean absolute percentage error
MBE mean bias error
MLP multilayer perceptron
MSE mean squared error
LSTM long-short-term memory
LUD Luchterduinen Wind Farm
NaN Not a Number
NN neural network
NRMSE normalized root-mean-square error
NWP numerical weather prediction
PACF partial autocorrelation function
PAWP Princess Amalia Wind Farm
PTU programme time unit
PV photovoltaics
RAM random-access memory
RANS Reynolds-averaged Navier–Stokes
RMSE root-mean-square error
RNN recurrent neural networks
RRV regel- en reservevermogen
RPF renewable power forecasting
SARIMA Seasonal Autoregressive Integrated Moving Average
SCADA supervisory control and data acquisition
SHAP Shapley Additive Explanations
SQL Structured Query Language
SVM support-vector machine
TSO transmission system operator
TU Delft Delft University of Technology
TVAL trade value
UST ultra-short-term
UST-WPF ultra-short-term wind power forecasting
UTC Coordinated Universal Time
VMD variational mode decomposition
VRE variable renewable energy
VWAP volume weighted average price
WT wavelet transform

WPF wind power forecasting

XBID European Cross-Border Intraday

XGBoost Extreme Gradient Boosting