論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Design of machine learning-based methods for wind speed and wind power forecasts
著者(和文)	Salazar Cueva Andres Aurelio
Author(English)	Andres Aurelio Salazar Cueva
出典(和文)	学位:博士(理学), 学位授与機関:東京工業大学, 報告番号:甲第12166号, 授与年月日:2022年9月22日, 学位の種別:課程博士, 審査員:肖 鋒,井上 裕嗣,青木 尊之,末包 哲也,大西 領
Citation(English)	Degree:Doctor (Science), Conferring organization: Tokyo Institute of Technology, Report number:甲第12166号, Conferred date:2022/9/22, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Design of machine learning-based methods for wind speed and wind power forecasts

Andrés Aurelio Salazar Cueva

Supervisors: **Prof. Feng Xiao**

Prof. Hirotsugu Inoue

Department of Mechanical Engineering

Tokyo Institute of Technology

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Science

June 2022

Abstract

To guarantee the reliability of a electrical grid that includes variable renewable resources (such as wind power), accurate power output estimations are necessary. However, regions with rugged terrain exhibit complex wind profiles and fluctuating patterns that make wind speed and wind power estimation challenging. On the one hand, numerical weather prediction (NWP) models cannot accurately capture local, close-to-surface patterns. On the other hand, pure data-based weather prediction cannot provide reliable long-term forecasts. Therefore, in order to successfully integrate wind power in existing electrical grids, methods to improve wind speed/power forecasts in zones with complex topography are necessary. This work explores machine learning-based methods for this purpose. First, we examine a machine learning model trained solely on past predictions and based on signal decomposition techniques and recurrent neural networks. Second, we introduce a multi-variable, deterministic neural network model inspired in denoising autoencoders, trained to clean the NWP forecast and produce a forecast with reduced bias. Finally, we present a probabilistic model that employs deep generative methods to produce probabilistic (ensemble) forecasts from the deterministic NWP result. We use observations from a real wind farms located in an areas with rugged terrain. The results of the present work show that (a) pure data-based methods are indeed not suitable for the aforementioned task, an (b) the proposed hybrid models that include both NWP and observations have superior skill when compared to other methods.

Acknowledgments

This work would not have been possible without the contribution, encouragement and support of many people.

First, I would like to express my most sincere appreciation and gratitude to my academic supervisor, Prof. Feng Xiao, for giving me opportunity of working under his supervision, for his patience and for his continuous guidance. Also, thanks to Prof. Hirotsugu Inoue for this support during the last year of my PhD.

My deep and sincere gratitude goes to Yuzhang Che for his support and contributions, his valuable insight into the wind power field and for providing me with useful tools that helped the development of this work. Also, thanks to CEF Co.,Ltd. for allowing the use of their data for this research.

To all the labmates in Xiao laboratory, thanks for always being willing to help and share your knowledge, as well as for providing a relaxing and comfortable study atmosphere.

The encouragement and moral support I received from my parents and my sister has been, despite the distance, a fundamental pillar in my daily life. Thank you for your unconditional love and for being always there for me.

I also want to thank all my friends for their support and companionship during these past two and a half years. To my friends in Japan: Xu, Elena, Erdem, Alfredo, Adrián, Luis, Konosuke, as well as to my beloved lifelong friends in Ecuador: Belén, Priscila, Pablo, Clara, thank you all.

Finally but not least, my deepest thanks to the Japanese Government for financially supporting my studies.

Contents

1 Introduction **11**

1.1 Overview of renewable energies 12

1.1.1 Major components 12

1.1.2 Integration into energy systems 13

1.2 Status of renewable energy in Japan 14

1.2.1 General situation 15

1.2.2 Wind power production and potential 16

1.3 Wind and wind power forecasting 17

1.3.1 The nature of wind 17

1.3.2 Meteorological models for wind speed forecasting 18

1.3.3 Extracting power for wind 20

1.3.4 Wind power estimation 20

1.4 Review of data-driven approaches for prediction and post-processing 23

1.4.1 General overview 23

1.4.2 Machine learning-based models 24

1.5 Main objective and outline of this thesis 26

1.6 Summary of this chapter 27

2 Fundamentals on machine learning **29**

2.1 Types of Machine Learning 31

2.2 Neural Networks 32

2.2.1 Long short-term memory networks (LSTM) 36

2.2.2 Deep generative models 38

2.3 Common practices in training neural networks 39

2.3.1 Dataset partitioning and hyperparameter tuning 40

2.3.2 Feature Scaling 41
2.3.3 Optimization Algorithm 42
2.3.4 Validation of models 43
2.4 Software 43
2.5 Summary of this chapter 44

3 Methodologies for the NWP and power models 46
3.1 Description of the wind farms of interest 46
3.1.1 Measurements datasets 48
3.2 The WRF model 52
3.2.1 Implementation in Awaji Island 53
3.2.2 Initial and boundary conditions 54
3.3 Wind power forecasts 54
3.3.1 Data preprocessing 55
3.3.2 Power-curve model 56
3.4 Evaluation tools 57
3.4.1 For deterministic models 57
3.4.2 For probabilistic models 59
3.5 Summary of this chapter 63

4 An LSTM model with variational mode decomposition 65
4.1 Description of the model 65
4.1.1 Variational mode decomposition 66
4.1.2 Architecture 72
4.2 Training details 72
4.3 Results 73
4.4 Discussion 81
4.5 Summary of this chapter 83

5 A deterministic multivariable neural network model 84
5.1 Description of the model 84
5.1.1 Denoising autoencoders 85
5.1.2 Architecture 88
5.2 Training details 92

5.3 Baseline method 92
5.4 Results 93
5.5 Discussion 103
5.5.1 Analysis of NN-SDPT 105
5.6 Tests during architecture design 106
5.7 Summary of this chapter 113

6 A deep generative model for probabilistic forecasting 114
6.1 Description of the model 114
6.1.1 The conditional variational autoencoder 116
6.1.2 Architecture 117
6.2 Training details 118
6.3 Baseline method 120
6.4 Results 121
6.4.1 As a deterministic forecast 122
6.4.2 As a probabilistic forecast 126
6.5 Power estimation 133
6.6 Discussion 136
6.7 Summary of this chapter 138

7 Conclusions and future work 140
7.1 Summary of this thesis 140
7.2 Conclusions 143
7.3 Final comments 144
7.4 Future work 145

List of Figures

1.1 Global renewable electricity generation 13
1.2 Electricity generation in Japan 15
1.3 Capacity and production of wind power in Japan 16

2.1 The machine learning approach 30
2.2 An individual neuron and a two-layer neural network. 33
2.3 Memory cell schematic 37
2.4 LSTM cell schematic 38
2.5 K-folds cross-validation schematic 43

3.1 Elevation map of Awaji Island 47
3.2 Elevation map of the Houhoku Area 48
3.3 Wind speed histogram (Awaji) 49
3.4 Wind speed histogram (Houhoku) 49
3.5 Average wind speed per hour of the day (Awaji) 49
3.6 Average wind speed per hour of the day (Houhoku) 50
3.7 Boxplots of wind speed values per site (Awaji) 50
3.8 Boxplots of wind speed values per site (Houhoku) 51
3.9 Observed data boxplots, per hour (Awaji) 51
3.10 Parent and nested WRF domains 53
3.11 Example of Tukey's method for outlier removal 55
3.12 Power curve fitting 56
3.13 Taylor diagram example 59
3.14 Schematic of the CRPS metric 60
3.15 Examples of rank histograms 62
3.16 Example of a dispersion diagram 63

4.1 VMD decomposition example 1 69
4.2 VMD decomposition example 2 70
4.3 VMD decomposition example 3 71
4.4 VMD+LSTM results, case 1 75
4.5 VMD+LSTM results, case 2 76
4.6 VMD+LSTM results, case 3 77
4.7 RMSE and MAE for all sites 78
4.8 Correlation coefficient for all sites 78
4.9 RMSE and MAE for all prediction times 79
4.10 Correlation coefficient for all prediction times 80

5.1 Example of a denoising autoencoder 86
5.2 The correction process of a denoising autoencoder 87
5.3 NN-S network 89
5.4 NN-SDPT network 91
5.5 Schematic of the MOS method 93
5.6 RMSE per site 95
5.7 RMSE per hour 99
5.8 Taylor diagram for all 15 sites accumulated 100
5.9 Taylor diagram for each of the 15 sites 101
5.10 Time series of observations and forecasts (corrected and uncorrected) 102
5.11 Analysis of the weights of NN-SDPT 105

6.1 Schematic of the conditional variational autoencoder (CVAE) . . . 118
6.2 Schematic of the analog ensemble method 121
6.3 RMSE and skill per site 122
6.4 RMSE and skill per hour 126
6.5 Continuous Ranked Probability Score (CRPS) and Reliability Index (RI) 127
6.6 Rank Histograms of wind speed forecasts 128
6.7 Dispersion diagrams of wind speed forecasts 129
6.8 Time series for wind speed and power. Case 1. 131
6.9 Time series for wind speed and power. Case 2. 132

6.10 Power curve for turbine No.2 134
6.11 Rank Histograms of the wind power 135
6.12 Dispersion diagrams of the wind power 135

List of Tables

3.1 Mesh resolution for each of the domains 54

4.1 RMSE, MAE and CC results per site 79
4.2 RMSE, MAE and CC results per hour 81

5.1 Description of the neural networks presented in this study. 92
5.2 RMSE values for all 15 sites 96
5.3 MAE values for all 15 sites 97
5.4 CC values for all 15 sites 98
5.5 Description of the selected case studies 103
5.6 Average RMSE and improvement for fully and locally-connected networks with 23 nodes in the middle layer 108
5.7 Average RMSE and improvement for fully and locally-connected networks with 24 nodes in the middle layer 109
5.8 Average improvement for locally-connected networks with 22, 23, 24 and 25 nodes in the middle layer 110
5.9 Average improvement for the NN-SDPT network and a similar network with one extra layer. 111
5.10 Average improvement for the NN-SDPT network similar networks with dropout layers 112

6.1 Description of the networks described in Figure 6.1 119
6.2 RMSE values for all sites 123
6.3 MAE values for all sites 124
6.4 CC values for all sites 125
6.5 CRPS and RI metrics 127

6.6 Energy and Variogram scores 130
6.7 RMSE, MAE, CRPS and RI for the power estimations 134

Chapter 1

Introduction

Climate change is a critical issue that threatens mankind as a whole. In 2017, the United Nations set as one of their Sustainable Development Goals to "Take urgent action to combat climate change and its impacts" [1], however, the global net emission of greenhouse gases (GHG) has kept on increasing and, during the decade of 2010-2019, GHG emissions were higher than any other previous time in human history [2]. Air-pollution alone causes between 4 to 7 premature deaths and hundreds of millions more are becoming ill [3]. Reversing and mitigating its effects require the involvement of diverse actors, from governments and policymakers to communities and individuals. Innovation and creation of new technologies is needed from most branches of science [2].

The largest source of CO_2 emissions is the global energy system. In order to limit warming below 2°C a radical transformation of the energy system over the next 30 years is needed; this includes reducing the use of fossil fuels and increasing the production from low- and zero-carbon energy sources [2].

This chapter summarizes the importance of renewable energies in achieving the aforementioned goal. This chapter starts with a brief description of the main sources of renewable energy and the challenges of integrating them into the electric grid. It continues into exploring the potential of wind energy globally and, in particular, for the case of Japan. The main motivation and objectives for this thesis are presented at the end of this chapter.

1.1 Overview of renewable energies

Many countries are already able to generate a considerable percentage of their electricity needs from renewable sources (RES). The four largest economies: the United States, Germany, China and Japan, generate between 20% and 30% of their electricity needs from RES. Canada, Brazil and New Zealand are among the countries currently able to generate between 60% and 80% from RES, while some few countries like Norway, Costa Rica, Paraguay and Nepal are close to 100 % [4].

This section provides a brief overview of the technologies that are driving the transition towards clean energy, as well as some of the challenges. The particular case of Japan is treated in Section 1.2

1.1.1 Major components

Figure 1.1 shows the main sources of renewable energy for the year 2020. By far, hydropower corresponds to more than half of all the generated renewable energy, followed by wind, photovoltaic solar, and biofuels. The availability and implementation of each resource varies by country and each presents both advantages and disadvantages. A brief introduction of each technology follows [5].

1. **Hydropower:** It is the world's largest source of renewable energy and it is expected to remain as so. Electricity can be harnessed from the flow of rivers or reservoirs; the latter can retain months or years of inflow and provides excellent flexibility.

2. **Wind power:** Electricity is harnessed from the movement of the wind. Onshore and offshore technologies have evolved rapidly and nowadays wind turbines have taller hub heights and larger rotor diameters. Usually, offshore wind turbines perform better due to better wind resources in the sea.

3. **Photovoltaic solar:** Converts sunlight into electricity. Solar panels can now be mass produced and their modularity allow a wide range of applications, from personal electronics to power generation facilities.

4. **Biofuels:** Fuel derived from biomass. These are considered a carbon-neutral source since burning biofuels will release the amount of CO_2 that was ab-

sorbed while the plant was growing. Common crops are soybean and palm oil trees [6].

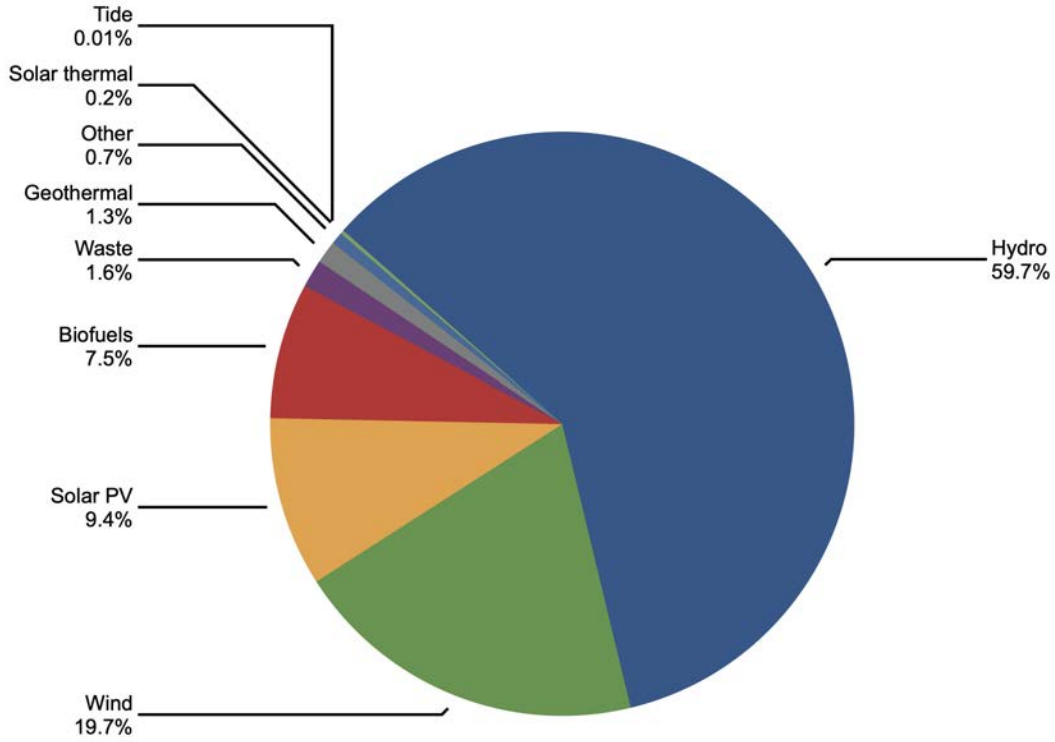

Figure 1.1: Global electricity generation by renewable source for 2020 [7]

1.1.2 Integration into energy systems

For renewable sources to replace conventional power plants successfully, they must be connected to the existing power transmission systems in a reliable way. This is of particular concern with solar and wind sources due to their dependence on the weather conditions, which normally results in intermittent availability. Integrating this type of *variable renewable energy* (VRE) sources into the grid presents many challenges, resulting in required adaptations that increase the complexity of the system. Some of these challenges can be classified into two groups:

1. **Grid stability** In order to guarantee continuous and uninterruptible power supply at a constant voltage and current, the stability, quality and reliability of the grid has to be maintained when VRE sources are online. This means control methods should be added to the system to ensure integration

requirements such as voltage ride through capability, injection or absorption of reactive current, frequency stability, active and reactive power control, harmonic distortion prevention, etc. Regulations vary by country and sometimes by grid company [8].

2. **Power balance** When several electricity sources (including VRE sources) are available, balancing load with generation results in a highly complex optimization problem. Most of the time preference is given to RES sources while backup generators must be ready to ramp up or down as needed [9]. Additionally, energy storage systems such as batteries may be necessary for storing energy surplus [8].This requires scheduling of each resource or *unit*, each of which has different flexibility levels, variability and constraints. The resulting problem is denominated the *unit commitment* problem.

Given these challenges, the National Renewable Energy Laboratory of the United States has identified four levels of flexibility that VRE power systems must address [10]:

1. **Flexible generation:** Efficient ramping up and down of power plants
2. **Flexible transmission:** Reduced bottlenecks and sufficient capacity for all resources
3. **Flexible demand-side resources:** Means to control the load directly or to respond changes in the market (smart-grids, energy storage, etc.)
4. **Flexible system operations:** Real-time decision making

It is clear by now that inaccurately predicting weather conditions and therefore incorrectly estimating the power output of VRE sources can not only impact the stability of the grid [8], but also interfere with decision-making and scheduling, possibly resulting in economic loss and increased CO_2 emissions.

1.2 Status of renewable energy in Japan

This section shifts our attention to Japan, which is the country of interest in this thesis. We start with a general overview of renewable energy in the country,

followed by a focused review on wind energy.

1.2.1 General situation

The energy situation in Japan has undergone rapid changes in the last decade, mostly sparked by the Fukushima Disaster of 2011. As an aftermath of the disaster, the nuclear power supply was drastically cut and a transition to renewable sources started gaining momentum [11, 12]. One of the latests developments was the announcement of the Japanese Government of an ambitious goal, to "make Japan a carbon-neutral, decarbonized society by 2050" [13].

Nevertheless, most of the electricity generated in Japan comes from non-renewable sources. Figure 1.2 shows the sources of all generated electricity in Japan for the year 2020. More than half the generated electricity came from fossil fuels, with the biggest sources being natural gas and coal. The third and fourth largest sources are hydropower and photovoltaic solar, the two most common RES in Japan.

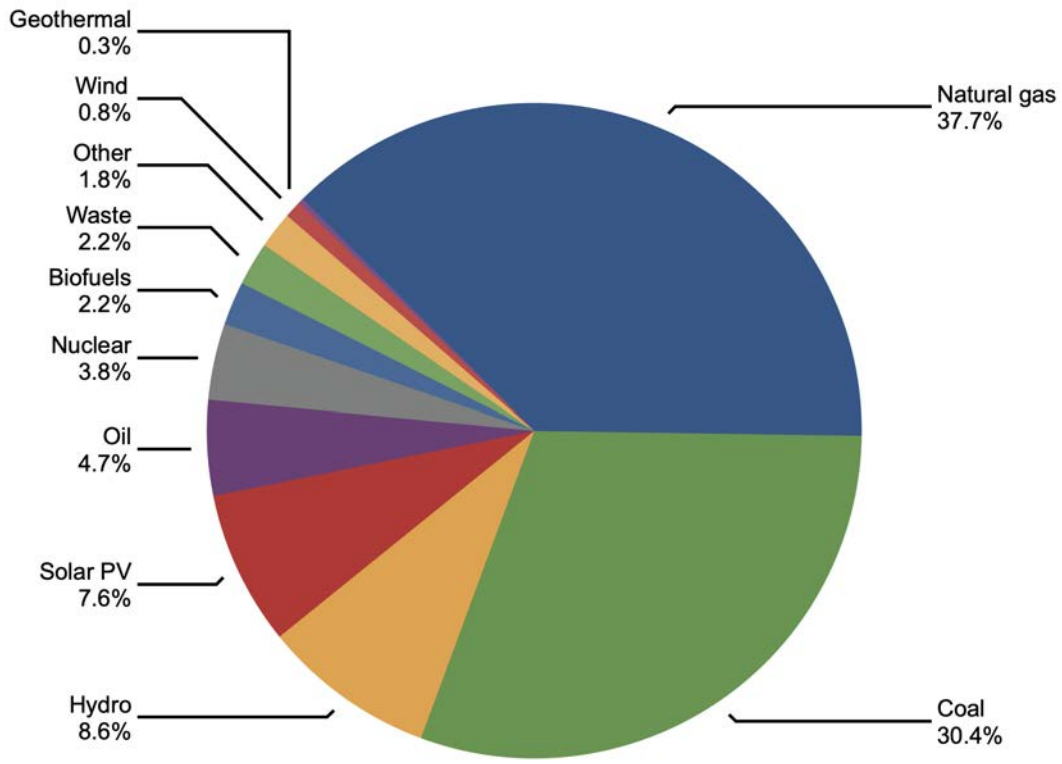

Figure 1.2: Electricity generation by source in Japan for 2020 [7]

1.2.2 Wind power production and potential

The Ministry of Environment of Japan has estimated in a 2011 report [14] an enormous wind power potential of as much as 1900GW. Production and generation have increased during the last decade; figure 1.2 shows two graphs, the upper panels displays the evolution of wind speed capacity in MW, and the lower panel displays evolution of wind speed production in GWh. Both have been steadily increasing, in 2020 Japan had the capacity to generate up to 4371 MW of wind energy, while in 2019 7673 MWh were actually generated.

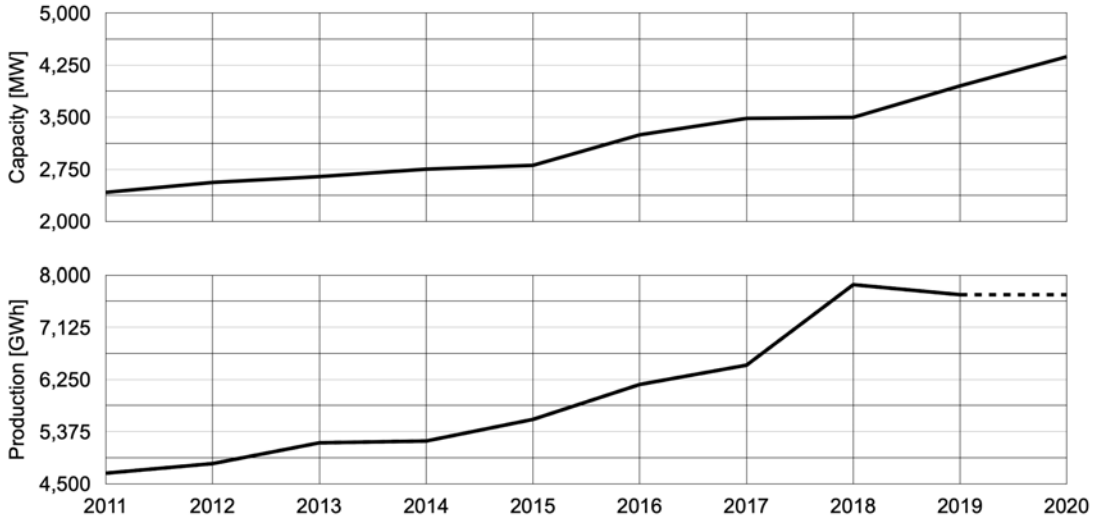

Figure 1.3: Capacity in MW and production in GWh of wind power in Japan 2011-2020 [15]

Despite the enormous potential, recent growth in capacity and favorable energy policies, the percentage of wind power contribution to the total generated electricity was about 0.8% in 2020 (figure 1.2). Several reasons for the slow adopting of wind power in Japan were given by Che (2017) [12], among these are Japan's high population density, social acceptance issues, complex terrain conditions and grid connection issues.

Complex terrain conditions is, in particular, an important issue when it comes to wind power generation. Understanding the wind profiles in zones with rugged terrain is necessary to estimate how much power can be obtained and achieve a smoother integration into existing power grids (section 1.1.2). The next sections in this chapter will explain how wind can be forecasted, and how power estimates

can be obtained from these forecasts.

1.3 Wind and wind power forecasting

This section will provide a short introduction into the RES of interest in this thesis: wind power. We start with a brief description of the natural phenomena of wind, and we continue with a review of the most common forecasting techniques. Finally we introduce the relationship between wind speed and power, and we provide a review of methods to estimate power output.

1.3.1 The nature of wind

Wind is the motion of air caused by differences in atmospheric pressure. Air flows form high pressure regions to the low pressure regions. Its speed is directly proportional to the pressure gradient, that is, the strongest winds occurs where the gradient is at its greatest. Wind flows in the three dimensions, however the vertical component is usually much smaller than the horizontal component. Vertically, warm air tends to rise and cool down, resulting in clouds and precipitation. Cool air tends to sink and warm up, resulting in evaporation of clouds and fair weather. Horizontally, in the northern hemisphere wind moves clockwise around high pressure zones and counter-clockwise around low pressure zones. The opposite happens in the southern hemisphere [16].

Three forces affect the behavior of wind:

1. **The pressure gradient force:** resulting from the difference in atmospheric pressure

2. **The Coriolis effect:** resulting from earth's rotation

3. **Friction:** resulting from the roughness of earth's surface

In general, the surface of the earth has great influence over the behavior of wind. Friction can not only slow wind down, but it can cause converging and/or diverging flows. In the case of rough geographic features (such as steep and mountainous terrain) turbulence and flow separation [17, 18] can also occur.

1.3.2 Meteorological models for wind speed forecasting

In its most basic form there are two main approaches for weather forecasting: *data-driven* and *dynamical* methods. Data-driven methods are empirical, i.e., they obtain predictions based solely on past observations. On the other hand, dynamical methods attempt to solve mathematical models that describe the state of the atmosphere as time progresses. A third category is that of *hybrid approaches*, which combines both methods in a variety of configurations and levels. All three alternatives will be briefly introduced. We also mention the approach of *ensemble forecasting* [19].

Data-driven models

Pure data-driven models operate based solely on past data, without the information from dynamical models. Within this category, *statistical forecasting* was the first forecasting technique to be used before dynamical models were introduced. Statistical models employ statistical tools to obtain predictions such as simple linear or non-linear regression. These models are also able to provide information of the uncertainty of the prediction (i.e., probabilistic forecasts) by using Bayesian statistics. Other data-driven frameworks include complex techniques such as signal decomposition [20], genetic algorithms [21], machine learning models such as neural networks [22], etc. Selecting adequate predictors is important in a data-driven model.

Dynamical models

These models are based on the physics and equations that govern atmospheric processes. These equations usually do not have closed-form solutions and discrete representations must be used. Normally the primitive equations (an approximation to the Navier-Stokes equations) are used. Space is also discretized, the horizontal dimension in longitude-latitude coordinates, and the vertical dimension in a coordinate which is function of hydrostatic pressure. All data and functions are represented as a finite set of grid points, for which the space and time derivatives are calculated by using numerical partial differential equation-solvers. Basic inputs to the model are initial and boundary conditions, topography data. More

advanced models are coupled with ocean circulation models and can even incorporate photochemistry schemes and biological processes. In general, these models tend to be complex and require large computational resources [23]. Since these models need numerical solvers are thus known as *numerical weather prediction (NWP)* models.

Hybrid approaches

The atmosphere is a non-linear dynamical system, which means that its state cannot be predicted in a deterministic manner. Therefore, dynamical and data-driven methods are commonly used in conjunction [24]. The NWP model provides a physical basis for the forecast, while a data-driven approach is used to post-process and enhance its results by incorporating information of past data. Hybrid models are of particular usefulness when close-to-surface predictions are needed. NWP models excel at resolving the state of the atmosphere at higher attitudes, but their reliability decrease once close to the surface [25]. On the other hand, data-driven methods are good at handling local weather and close-to-surface patterns [19, 24, 26].

Ensemble forecasting

Ensemble forecasts were introduced by Leith (1964) [27] as a way to obtain information about the stochastic dynamics and uncertainty of a deterministic prediction by using Monte Carlo methods. It starts by defining an ensemble of initial conditions, i.e., a finite sample drawn from distribution that describes the uncertainty of the initial state of the atmosphere. A dynamical NWP forecast is repeatedly run for each sampled initial state evolving them to a future time. The dispersion of each ensemble member after simulation is considered as indicative of the uncertainty of the forecast. Their accuracy and reliability is directly proportional to the number of ensemble members, therefore ensemble forecasts are known to provide excellent results, however, at a high computational cost.

1.3.3 Extracting power for wind

Power can be obtained from the movement of wind by using wind turbines. Average turbine generators range from 300 to 600 kW while newer models range between 1 to 3 MW. Advancements in many areas of engineering have played an important part in making wind power affordable, and now wind energy is one of the power sources with lowest cost. Among the main factors for the development of wind power technology are advances in material science, power electronics and computer simulation and prototyping [28].

Theoretically, the power output P_o from a turbine is directly proportional to the cube of the wind speed u, as given by the following equation [29].

$$P_o = \frac{1}{2}\rho C_p A u^3 \quad [\mathrm{W}] \tag{1.1}$$

where ρ is the air density in $\mathrm{Kg/m^3}$, A is the wind turbine rotor swept area in $\mathrm{m^2}$ and C_p is the dimensionless power coefficient. This coefficient depends on turbine specifications such as the tip speed ratio, and blade pitch angle.

In practice however, the ability of a turbine to extract power depends on the following factors according to [28]:

1. Wind power availability
2. Power curve of the turbine
3. Ability of the machine to respond to wind perturbations

Therefore, to maximize power extraction in a location each of these factors has to be studied. In this work we address the first two, forecast models of wind availability were reviewed in section 1.3.2, while the calculation of the power curve is discussed in the following section.

1.3.4 Wind power estimation

While it is possible to employ equation 1.1 to obtain an estimation of wind power given the wind speed, the actual generated power from a turbine depends on countless factors specific to each turbine's mechanical characteristics: rotor, generator, gearbox and efficiencies of the components, as well as the predetermined

cut-in, cut-out and rated speeds [30]. It is common for manufacturers to provide power curves that take into account these technical details, however several factors such as site-specific wind dynamics, wear and tear of the turbine, electrical losses, topographical effects, wind farm layout, etc., introduce discrepancies that make these approximations inadequate models to estimate wind power [31]. For this reason, statistical analysis of data is the current foundation for most modelling techniques. A review on these techniques is presented in this section [30].

Parametric methods

The power output of a turbine is usually modelled as:

$$P_o(u) = \begin{cases} 0 & \text{if} \quad u < u_c \text{ or } u > u_s \\ p(u) & \text{if} \quad u_c \leq u \leq u_r \\ P_r & \text{if} \quad u_r \leq u \leq u_s \end{cases} \tag{1.2}$$

where u_c and u_s are the cut-in and cut-out speeds respectively, u_r is the rated speed, $p(u)$ is the non-linear relationship between power and speed, and P_r is the rated power. Parametric models attempt to find mathematical expressions for these relationships. Some examples are:

1. **Linearized segmented models** [32, 33, 34]: Each of the segments is approximated by a linear equation and fitted through least square methods.

2. **Polynomial power curve models** [35, 36, 37]: These models attempt to approximate the non-linear segment $p(u)$, often resulting in a better fit. This category includes quadratic, cubic and ninth-degree polynomials, as well as exponential power curves.

3. **Maximum principle method** [38]: Employs wind speed and power bins and the density of the points in each bin to obtain an estimate.

4. **Dynamical power curve** [38]: The dynamics are separated into a deterministic and a stochastic part, assuming that the stochastic portion satisfies the Markovian property. According to [30], this method yields satisfactory machine-specific and site-independent estimations.

5. **Probabilistic models** [39]: These models estimate the uncertainty in wind power output assuming it follows a normal distribution with varying mean but constant variance.

6. **Logistic functions** [40, 41, 32]: Since the shape of the power curve is similar to that of a logistic function, four parameter and five parameter logistic function approximations have been applied.

Non-parametric methods

Non-parametric models attempt to find a solution for the equation

$$P = f(u) \tag{1.3}$$

This category includes copula models, spline interpolation, neural networks, fuzzy methods, data mining algorithms, among others.

The power estimation methods shown in this section are fitted by using datasets of speed and power for particular turbines and locations. However, the base relationship between wind speed and power (shown in equation 1.1) is nonetheless true for all cases: power is proportional to the cube of wind speed. This means that a small perturbance in wind speed will transform into a power perturbance proportional to the cube of the original speed perturbance. In terms of power forecasting, and as it will be seen in the following chapters, an error of just a few meters per second in the wind speed forecast can translate into a miscalculation of hundreds of kilowatts in the power forecast. For this reason, accurately predicting wind speed is crucial for power estimations.

In this work we are interested in employing a data-driven approach to obtain enhanced wind speed forecasts that can, in turn, be employed to obtain power estimations. The next section provides a literature review on general and data-based methods with a focus on machine learning. A proper introduction to the machine learning methods used in this work is given in chapter 2.

1.4 Review of data-driven approaches for prediction and post-processing

In this section we review some data-driven approaches for prediction or enhancement of weather forecasts. We include both pure data-driven approaches and hybrid implementations in conjunction with an NWP model. At the end of this section we focus on those methods that employ machine learning tools.

1.4.1 General overview

Classic data-driven approaches correspond to those statistical methods based on regression. Simple and multiple linear regression are commonly used although these require assumptions on the distribution of the data and the residuals (such as Gaussian residuals with constant variance). When these assumptions are untenable, nonlinear regression can be used. Generalized linear models and Poisson regression are two nonlinear regression commonly used in the atmospheric sciences [19].

As mentioned in section 1.3.2, statistical methods are also in a hybrid manner for the post-processing of NWP forecasts. Among the most well-known are the Kalman filter [42] and its variations, Model Output Statistics (MOS) [43], Ensemble Model Output Statistics (EMOS) [44], and quantile regression [45].

A particular post-processing method of interest for this work is the generation of probabilistic forecasts from deterministic forecasts. Some existing post-processing approaches allow us to obtain probabilistic ensemble forecasts without the need to repeatedly run the dynamical model. Some of this methods include Regression Estimation of Event Probabilities (REEP) [46], logistic regression [47, 48] and analog schemes [49, 50]. This last approach generates ensembles by using past analog pairs of observations and forecasts, and has been shown to produce statistically consistent ensembles for both wind and temperature.

1.4.2 Machine learning-based models

In recent years, machine learning (ML) and artificial intelligence (AI) techniques have also been developed, due to the abundance of forecasted and observed data.

Traditional post-processing methods have been improved by the inclusion of ML components. In the context of ensemble forecasts, Rasp and Lerch [51] replaced the link functions in a distributional regression model with a neural network able to predict the distribution parameters from a set of predictors, and applied it to 2-m temperature forecasts. The quantile regression method, originally developed by Bremnes [45], was enhanced by Cannon [52] with the development of a monotone composite quantile regression neural network; and again by Bremnes [53] by using neural networks to estimate the coefficients of Bernstein basis polynomials, after which the quantile function is constructed. On the other hand, Taillardat et al. [54] developed instead a random forest generalization for quantile regression, first as a non-parametric approach and posteriorly extended within a semiparametric model [55]. Moreover, Chapman et al. [56] introduces the use of convolutional neural networks for the correction of gridded, two-dimensional integrated vapor transport (IVT) forecasts, based on image processing techniques and denoising autoencoders. Regarding the post-processing of ensemble forecasts, a review is given in Vannitsem [57].

The probabilistic forecast generation task has also been tackled with machine learning methods. Scher et. al. [58] used convolutional neural networks trained on past weather forecast to obtain confidence values of medium-range forecasts. Du [59] has employed a collection of forecasts and observations to train several machine learning algorithms, and construct an ensemble forecast with the outputs of each ML algorithm as members.

A rapidly developing branch of data science and machine learning is that of generative models, where the goal is to learn the underlying probability distribution of the data in order to be able to draw new samples from it. A notable attempt of using generative models to generate probabilistic forecasts is that of Fanfarillo et. al. [60], where one class of generative models, the conditional variational autoencoder (CVAE), was trained with historical forecasts and observations in order to generate analog ensembles. CVAE is based on stochastic variational

Bayes and is one of several of recently developed deep conditional generative models. Generative models are discussed more in depth in section 2.2.2.

Wind speed related applications

For the particular case of wind speed, most ML-related applications have been on prediction based solely on past time series. A common approach would consist of a regression ML model [61, 26, 62, 63], often in combination with an error correction or optimization component [64, 65]. These implementations lack an NWP component, and most of literature is focused to predicting just a few time steps ahead.

On the other hand, attempts at employing ML for error correction include the use of nonlinear autoregressive exogenous (NARX) networks [66], Markov stochastic processes [67], sequence transfer algorithms [68], polynomial neural networks [24], support vector machines [69], and bidirectional gated recurrent unit (GRU) neural networks [70]. All of these methods require the output from an NWP model in either deterministic or ensemble form.

It should be mentioned that very few implementations of NWP post-processing methods are targeted at power estimation. On one hand, generative models have been extensively used for *renewable scenario generation*, that is, generating renewable energy production scenarios for power system planning and operation. On the other hand, most implementations rely solely on historical data. Including NWP forecasts in the models seems to be a rather unusual practice. Comprehensive reviews can be found in Khoshrou and Pauwels [71] and Mashlakov at al. [72]. Dumas et al. [73] is one of the few studies where both weather and power data were used to obtain power predictions, however, in a direct manner without obtaining intermediate improved weather information.

To the best of the authors' knowledge, none of these methods have been assessed taking into account the main criteria of interest in this work: wind power estimation in an area with remarkably fluctuating wind speed dynamics due to complex topography. Therefore, the creation and development of such models is imperative.

1.5 Main objective and outline of this thesis

The previous section showed that literature regarding wind power estimation in zones with rugged terrain is insufficient. We are therefore interested in developing models that adjust to this criteria.

Broadly speaking, there are two main approaches for wind power estimation [74, 75]. The *direct approach* uses past power data and obtains predictions based on those past time series. Methods for time series forecasting such as the *autoregressive integrated moving average (ARIMA)* are often used. On the other hand, the *indirect approach* obtains power estimation from a wind speed forecast and one of the power curve models described in section 1.3.4.

It is in the context of the indirect approach that the main goal of this thesis is defined. We propose to improve the wind speed forecasts used for power estimation by using machine learning models, with a focus on zones with rugged terrain and therefore complex wind profiles. The variety of machine learning models let us tackle this problem from many different viewpoints. In this work, we will specifically explore the following approaches:

- A wind speed prediction model based solely on past observations.
- A *deterministic* correction model to obtain enhanced NWP wind speed predictions, by using past observations.
- A *probabilistic* correction model to obtain enhanced NWP wind speed predictions, by using past observations.

All models are fitted by using observations from two wind farms located in zones with rugged terrain, and configured to produce 24-hour day-ahead wind speed forecasts. The correction models include also NWP forecasts for the same location. We expect the machine learning models to capture the local patterns and close-to-surface patterns caused by the rugged terrain. With respect to the correction models, we propose a deterministic model and a probabilistic model in order to highlight the importance of uncertainty in wind speed forecasts. The deterministic model is evaluated on its skill to produce enhanced forecasts (reduced

error), while the probabilistic model is evaluated for both, wind speed and wind power estimation.

Given the goal of this thesis, the outline is as follows.

- Chapter 2 will introduce the basics of machine learning and, specifically, of neural networks, which are the main ML model to be used in this work. Also, the process of training a neural network, common practices, LSTM networks and generative models are also described here.

- Chapter 3 describes the methodology, including the wind farms of interest located in zones of fluctuating wind due to complex topography. The NWP model and the power curve model are also discussed. The tools and methods to evaluate the models are also in this chapter.

- Chapter 4 presents the wind speed prediction model based solely on past data (pure empirical model). We employ signal decomposition techniques and time series-focused machine learning methods.

- Chapter 5 introduces the deterministic wind speed correction model, which was inspired by denoising autoencoders and incorporates a multi-variable input.

- Chapter 6 introduces the probabilistic wind speed correction model, which is based on neural network generative models, and it is evaluated for actual wind power estimation.

- Chapter 7 provides the overall conclusions and comments.

1.6 Summary of this chapter

In this chapter the problematic regarding climate change and the need for a transition to renewable energy generation was discussed. It was seen that globally the transition is on its way, with four technologies leading the way: hydropower, wind power, photovoltaic solar and biofuels. We highlighted that in the case of Japan the transition has been slow when compared to other countries, despite

having considerable potential for renewable energy generation, and wind power in particular.

The importance of power estimation in terms of grid connectivity and reliability means that obtaining accurate predictions for variable renewable resources such as solar and wind is fundamental and necessary. In the case of Japan, obtaining wind forecasts is complicated by the complex mountainous terrain of the country. A literature review revealed that methods for improving weather forecasts targeted at power estimation in zones with rugged terrain is virtually non-existing.

Given this context the main goal of this work was hence established. It is of interest to develop models able to produce accurate wind speed forecasts in areas of rugged terrain, focused on power estimation. Machine learning approaches were chosen due to the availability of both observed and predicted data. This data comes from a wind farm located in a zone with mountainous and steep terrain, therefore providing the model information about complex and turbulent wind profiles. Overall three models will be introduced, one prediction model based on past observations, and two correction models based on past observations and NWP forecasts.

Chapter 2

Fundamentals on machine learning

During the last years advances in artificial intelligence (AI) and machine learning (ML) have demonstrated that they are capable of astonishing achievements. From real-time translation to improvements in cancer detection and prediction of stock markets, AI and ML are being adopted by business and enterprises at an accelerating pace due to their versatility and adaptability.

When tackling a problem, traditional programming techniques require the writing of an algorithm with instructions describing how to solve the problem. For trivial problems, algorithms can be simple. However, for any new iteration of the problem, the algorithm must be updated. Over time it could become hard to maintain. The fundamental difference with a machine learning approach is that the problem-solving algorithm is replaced with a model able to learn from data, as shown in figure 2.1. If the program needs to solve a new variation of the problem, the model can learn from incoming fresh data. For this reason, it is said that *machine learning is the field of study that gives computers the ability to learn without being explicitly programmed* [76].

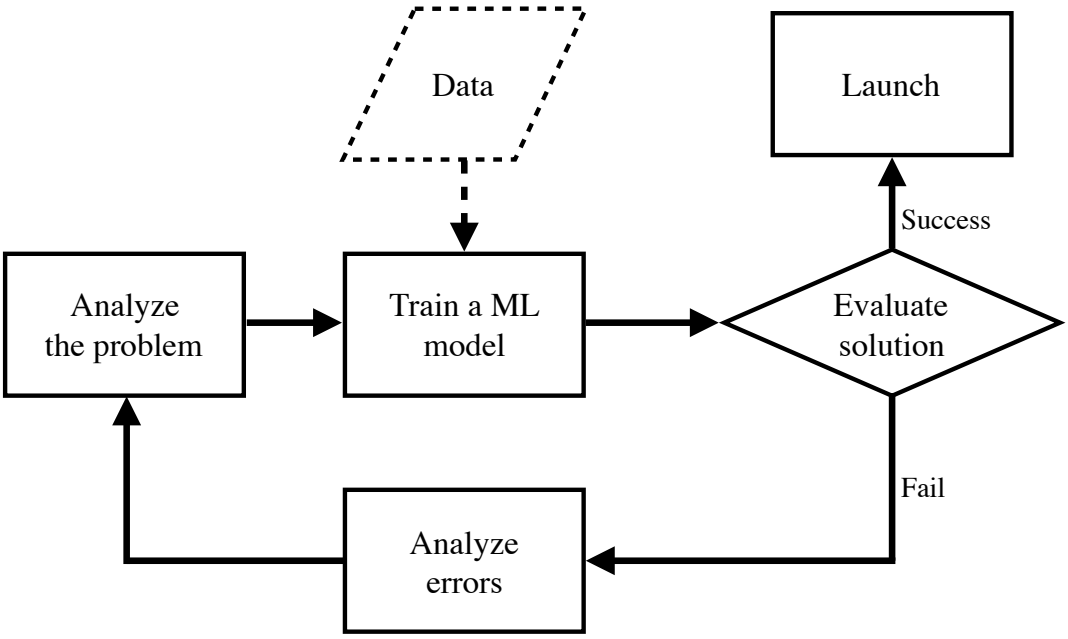

Figure 2.1: The machine learning approach (adapted from [76]).

A formal definition of this process is given by Goodfellow [77]: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E". The experience E comes from the data set. An appropriate data set contains a number of *examples*, each of which consists of a collection of features that have been quantitatively measured from the environment of interest. One of these examples is denoted as a vector $\mathbf{x} \in \mathbb{R}^n$ where each entry x_i is a feature.

The performance measure P is related to the selected task T. As mentioned before, the kind of tasks T that machine learning excels at solving are those for which writing fixed algorithms becomes impractical. Common applications range among classification, regression, transcription, translation, anomaly detection, denoising, among others [77].

2.1 Types of Machine Learning

Depending on the characteristics of the dataset and the final purpose, machine learning implementations can be classified in three broad, non exclusive categories [76, 77].

1. **Supervised learning:** In a supervised learning task, each example $\mathbf{x}$ is associated with a label y. The learning process involves observing both, the examples $\mathbf{x}$ and the labels y in a data set, then learning to predict y from $\mathbf{x}$ (this means, estimate the probability distribution $p(y \mid \mathbf{x})$).

2. **Unsupervised learning:** In an unsupervised learning task, the examples $\mathbf{x}$ do not have an associated label. The learning process consists of estimating the probability distribution $p(\mathbf{x})$ to find interesting properties or patterns in the data.

3. **Reinforcement learning:** A reinforcement learning task does not require a fixed data set, but rather an observable and interactive environment. The learning process consists of constantly interacting with the environment in order to obtain experience E, and learn the best strategy or policy to solve a task T within the environment.

Learning algorithms

In this short section we briefly introduce the most common machine learning algorithms. A broader review is presented in Geron [76] and Goodfellow [77].

- **k-Nearest Neighbors:** This non-parametric model scans for the K nearest points of a test input $\mathbf{x}$ in the feature space. The test input is assigned to the most common class among this set. The nature of the algorithm makes it unpractical with high-dimensional inputs. The more features in the input vector $\mathbf{x}$, the higher the dimensionality of the feature space. This makes the examples to become sparser, affecting the performance of the models since the "neighbors" are no longer close.

- **Support Vector Machines (SVMs):** SVMs also resort to the feature space to make predictions. Training a SVM involves finding hyperplanes in

the feature space, creating partitions where all or most of the examples in one partition correspond to a certain class. When a new input $\mathbf{x}$ is given, the class is assigned based on the partition where $\mathbf{x}$ falls in the feature space. In most applications, the classes in the feature space will not be able to be separated with a hyperplane (linearly separable), so transformations to the data such as the creation of polynomial or features, normalization and dimensionality reduction, are often necessary.

- **Decision Trees and Random Forests:** Decision trees also work by partitioning the feature space. However, the main difference with SVMs is that decision trees recursively partition the space in hyper-rectangles. The recursive nature of the partitioning results in a tree-like decision process that is followed every time a prediction need to be made. The boundaries of the partitioning are set by minimizing a predetermined cost function. A random forest consists of the simultaneous implementation of several decision trees.

- **Artificial Neural Networks (ANNs):** ANNs are a radical departure from the previously mentioned methods. Neural networks consists of layers of *artificial neurons* developed based on real biological neurons. The neurons are connected by weights and subjected to a nonlinear function. The network is constructed by stacking neurons in layers. Neural networks are currently used in a variety of applications, from computer vision to speech recognition.

This research uses ANNs as the main approach to develop the proposed models. In the following sections we describe the basic workings of an artificial neuron and a neural network. We also introduce two approaches that use complex implementations of neural networks: the long short-term memory (LSTM) network and deep generative methods.

2.2 Neural Networks

The basic unit in a neural network is called the *neuron* or a *node*. As shown in figure 2.2, a node receives several inputs x_1, x_2, $\cdots$ and each is associated with a

weight w_1, w_2, $\cdots$. The output y of the neuron consists of a non-linear function h, called the *activation function*, applied to the weighted sum of the inputs.

$$y = h\left(\sum_i w_i x_i\right) = h\left(\mathbf{x}^T \mathbf{w}\right) \tag{2.1}$$

In practice, the most commonly used activation functions are

- **Sigmoid function:**

$$\phi(x) = \frac{1}{1 + e^{-x}} \tag{2.2}$$

- **Hyperbolic tangent:**

$$\tanh(x) = 2\phi\left(2x\right) - 1 \tag{2.3}$$

- **Rectified Linear Unit (ReLU):**

$$\mathrm{ReLU}(x) = \max(0, x) \tag{2.4}$$

Stacking several nodes together results in a *layer*, and interconnecting several layers together creates a neural network. An example of a simple neural network is also shown in figure 2.2, consisting of three main parts: an input layer, two hidden layers and an output layer. In this kind of network, information only moves in one direction, and there are no feedback connections such as cycles or loops. For this reason, these networks are called *feedforward neural networks* [77].

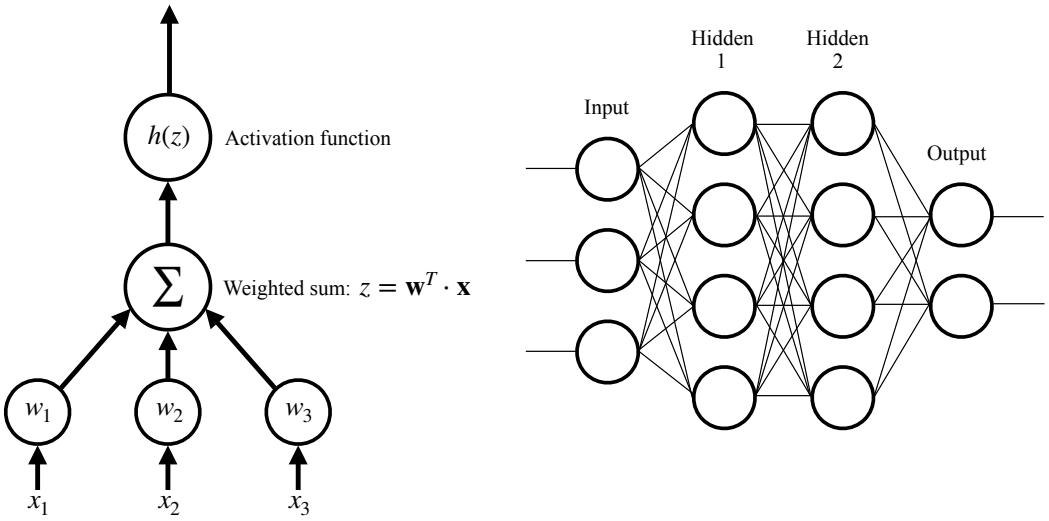

Figure 2.2: An individual neuron and a two-layer neural network.

If between any two layers all nodes are connected to each other (as shown in fig 2.2), we call those layers to be *fully-connected* or *dense*. Networks where all layers are fully-connected can also be called *fully-connected networks* or *dense neural networks (DNNs)*.

Each of the three sections in a network has different characteristics:

1. The input layer consists of *passthrough* nodes, meaning that their output will be the same as their input. The number of nodes in this layer must be the same as the size of the vectors $\mathbf{x}$ in the data set, plus one node with a constant bias value of 1.

2. The hidden layers perform the computations described in equation 2.1. A network with more than one hidden layer is called *deep feedforward network*. There is no deterministic method to find an optimal number of hidden layers in a network, nor optimal number of the nodes in the hidden layers. Network design is often based on trial and error.

3. The output layer performs the same computation as in 2.1. However, depending on the application, each individual activation function could be replaced by a shared activation function. The necessary number of nodes in the output layer depends on the application as well.

With this configuration, it can be shown that a neural network is a universal approximator as long as enough nodes are available [76].

Training a neural network consists on optimizing all weights in all nodes (the parameters of the network Θ) so that for each training example $(\mathbf{x}, y)$, the output of the network $\hat{y}$ resembles y as much as possible. Under this context, we call y our *target*. The training process can be roughly summarized in three steps:

1. Initialize the parameters of the network.

2. Feed all or part of the training data as input. Obtain a preliminary output $\hat{y}$

3. Given a cost or loss function J that provides a disimilarity measure, calculate its value for y and $\hat{y}$. One of the most commonly used loss function is the mean square error (MSE):

$$J(\Theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(y_{(i)} - \hat{y}_{(i)} \right)^2 \tag{2.5}$$

4. Measure the contribution of the each connection to the error. In order to find the change in J with respect to the weight $\Theta_{jk}^{(l)}$, that is, the weight between node j in layer $l+1$ and node k in layer l, we calculate

$$\frac{\partial J}{\partial \Theta_{jk}^{(l)}} = a_k^{(l)} \delta_j^{(l+1)} \tag{2.6}$$

where $a_k^{(l)}$ is the output of node k in layer l and

$$\delta_j^{(l+1)} = \frac{\partial J}{\partial z_j^{(l+1)}} \tag{2.7}$$

In order to find 2.6 for all parameters the calculation has to start from the output nodes and flow towards the input nodes. Therefore, this algorithm is called the *backpropagation algorithm* [78].

5. Update each weight in proportion to its contribution to the error, by using an optimization algorithm. For example, when using gradient descent, the update can be written as:

$$\Theta_{jk}^{(l)} := \Theta_{jk}^{(l)} - \alpha \frac{\partial J}{\partial \Theta_{jk}^{(l)}} \tag{2.8}$$

where α is the learning rate. In general form, it is said that the parameters of the network are updated by subtracting the gradient with respect to the parameters $\nabla_\Theta J(\Theta)$ multiplied by the learning rate:

$$\Theta \leftarrow \Theta - \alpha \nabla_\Theta J(\Theta) \tag{2.9}$$

Steps 2-4 are repeated until convergence. The learning rate is an additional parameter that need to be tuned in order to achieve proper convergence. Since this parameter is different from the parameters (weights) of the network, it is

referred as a *hyperparameter*. More complex models can have many hyperparameters that require additional tuning, and property tuning these hyperparameters is an important part in the training process.

After some repetitions of the procedure detailed above the network will have seen at some point all of the training data. This is called an *epoch*. Training is usually described in terms of epochs, for example, training a model for 20 epochs means that the same data has been fed to the model 20 times.

The remaining of this section will focus on two machine learning frameworks that use neural networks as their foundation. These two frameworks will be used in later chapters of this work.

2.2.1 Long short-term memory networks (LSTM)

The neural network shown in figure 2.2 is feedforward, meaning that the connections do not point backwards and do not form loops. A different type of network is the *recurrent neural network (RNN)*, which contains recurrent neurons or layers. This type of networks are design to be used with sequential data, such as time series, trajectories, or text.

The way these recurrent layers handle sequential data is by receiving the input sequentially. At each time step t, the layer receives the corresponding input for time t as well as an additional input generated at the previous $t-1$. This additional input is called the *hidden state* $\mathbf{h}$ and it contains information about previous time steps. In some simple RNNs, the hidden state corresponds simply to the output from the previous time step (that is, $\mathbf{h}_{t-1} = \mathbf{y}_{t-1}$), while in more complex models it has other interpretations. Since this vector stores information from past inputs, it is said that the neurons or layers have a form of memory. Therefore, in the context of RNNs, a collection of recurrent neurons or layers is referred as a *memory cell* [76]. Figure 2.3 shown a schematic of a memory cell, with the process represented against the time axis. Note that it is the same cell represented once per time step.

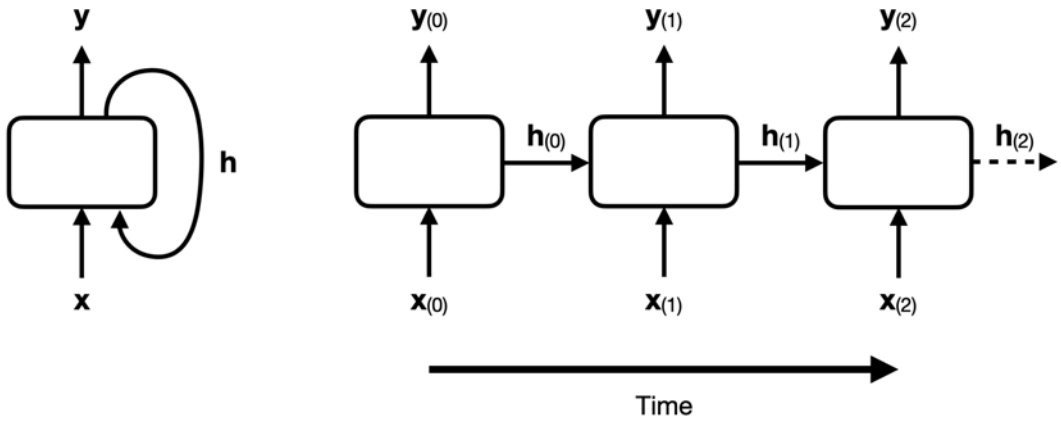

Figure 2.3: A memory cell schematic with an input $\mathbf{x}$, output $\mathbf{y}$ and hidden state vector $\mathbf{h}$ (adapted from [76]).

In this work we employ a more complex type of memory cell called the long short-term memory (LSTM) cell [79]. An schematic is shown in figure 2.4. Each box inside the cell represent a fully-connected layer with their respective activation function (shown in pink for the logistic function and in yellow for the hyperbolic tangent). In this cell the state is split into two vectors: a short-term state $\mathbf{h}_t$ and a long-term state $\mathbf{c}_t$. $\otimes$ represents element-wise multiplication and $\oplus$ represents element-wise sum. For every time step t, the input $\mathbf{x}_t$ and the short-term state from the previous time $\mathbf{h}_{t-1}$ are fed to the four layers. From left to right:

1. The first layer outputs $\mathbf{f}_t$. This output is multiplied with the long-term state $\mathbf{c}_t$ at what is called the *forget gate*. Therefore, $\mathbf{f}_t$ controls how much of $\mathbf{c}_t$ is erased at each time step.

2. The second layer outputs $\mathbf{g}_t$. This is the main layer and it analyzes and returns information from both $\mathbf{x}_t$ and $\mathbf{h}_{t-1}$.

3. The third layer outputs $\mathbf{i}_t$. This vector is multiplied with outputs $\mathbf{g}_t$ at what is called the *input gate*. Therefore, $\mathbf{i}_t$ controls how much of outputs $\mathbf{g}_t$ should be kept.

4. The fourth layer produces $\mathbf{o}_t$. As seen in the schematic, this vector is multiplied with the output of the standalone hyperbolic tangent function at the

output gate and results in the final outputs $\mathbf{y}_t$ and $\mathbf{h}_t$. It therefore controls which part of the long-term vector should be kept.

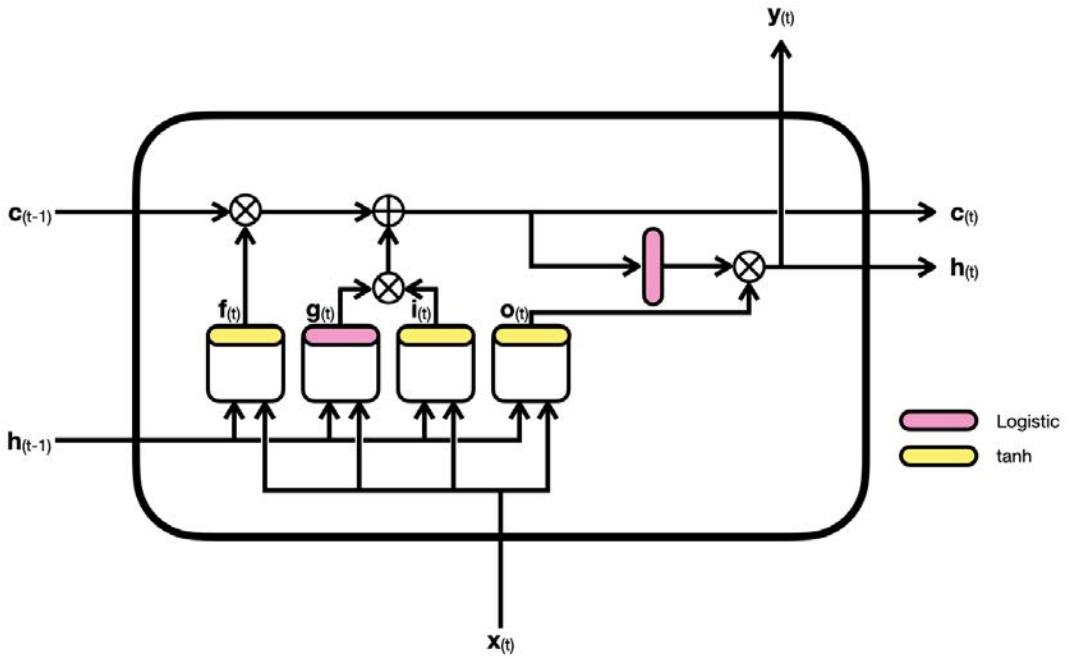

Figure 2.4: An schematic of a LSTM cell (adapted from [76]).

2.2.2 Deep generative models

Within the many ways to classify machine learning techniques, one of the most important is arguably *discriminative* and *generative* modeling. Here we understand *modeling* as devising a model for a phenomenon, able to accept an input, simulate the behavior of the phenomenon, and be evaluated through observations. Under this context, the purpose of a discriminative model is to learn a predictor given the observations. On the other hand, the purpose of a generative model is to learn the joint distribution over all variables. That is, a generative model attempts to learn how the data is generated in the real word. Deep generative models are, therefore, generative models that employ neural networks with multiple layers to learn a probability distribution. This section will provide a short insight into this type of machine learning models [80].

Assume that $\mathbf{y}$ is a vector representing the observed variable whose joint distribution we want to model, taken randomly from an unknown underlying process.

We denote the true probability of $\mathbf{y}$ as $p^*(\mathbf{y})$. Then, the task of interest is to learn the probability distribution $p_\theta(\mathbf{y})$ with parameters θ:

$$\mathbf{y} \sim p_\theta(\mathbf{y}) \quad \text{where} \quad p_\theta(\mathbf{y}) \approx p^*(\mathbf{y}) \tag{2.10}$$

That is, we are interested in finding the parameters θ such that $p_\theta(\mathbf{y})$ approximates the true distribution $p^*(\mathbf{y})$ for any $\mathbf{y}$.

However, as mentioned before, the models we are interested in require an input or a condition. In that case, we are interested in the conditional distribution over the values of $\mathbf{y}$ conditioned on the values of the input $\mathbf{x}$, and we must find the parameters θ such that

$$p_\theta(\mathbf{y}|\mathbf{x}) \approx p^*(\mathbf{y}|\mathbf{x}) \tag{2.11}$$

The capability of neural networks to be function approximators allows them to be used in this type of probabilistic models as well. Neural networks can be used to parametrize probability density functions (PDFs) or probability mass function (PMFs) in probabilistic models. Also, neural networks are scalable and allow stochastic gradient based optimization (section 2.3.3), and this makes them attractive for parametrization of PDFs derived from larges models and large datasets.

Two popular frameworks for generative models that involve neural networks are generative adversarial networks introduced by Goodfellow et al. [81], and the Autoencoding Variational Bayes model introduced by Kingma and Welling [82], also known as variational autoencoders. In this work we will employ a variation of the latter to develop our probabilistic models in chapter 6.

2.3 Common practices in training neural networks

To train the neural networks we will follow the practices outlined in this section, when applicable. These practices are common in the machine learning field and following them help obtain statistically valid models and avoid data leaks. These range from the preprocessing of data to the validation of different trained models.

2.3.1 Dataset partitioning and hyperparameter tuning

It is of interest to obtain models able to give accurate predictions when new data is available, that is, we want model to be able to *generalize* to new cases. In order to test how well a model generalizes, it is necessary to partition the available dataset into smaller sets. A percentage of the data is used during training, and the remaining, unseen data is used to evaluate the performance of the model with unseen cases.

The most common partition scheme is into three sets: a training set, a validation set and a test set. This three-set splitting strategy is useful especially when comparing between to models, or when comparing two instances of the same model but trained with different hyperparameters. For example, a particular neural network can be trained with different learning rates and later compared to discover the most effective value for learning rate hyperparameter. This is called *hyperparameter tuning*. The training and evaluation process is as follows [76].

The validation set is not only useful for hyperparameter tuning, but also to determine when to stop training. The *validation error* is the error of the model on the validation set, and it is calculated after every epoch. It is common to monitor the validation error during training and to stop the process once it stops decreasing. This is called *early-stopping*. In some cases however it is good practice to let training continue, in case a lower local minimum is found. This is called *patience*, and the *patience hyperparameter* is the number of epochs that the model runs after having found a minimum validation error.

1. Train the model of interest using the training set and an initial collection of hyperparameters.

2. Evaluate the model with the unseen data from the validation set, using a metric of interest. It is common to employ the same loss function as an evaluation metric since it is the function being optimized.

3. Modify one or more hyperparameters and repeat step 1 and 2, with the goal of obtaining better metrics (lower error) each time. Choose the model with the best metric.

4. Evaluate the chosen model with the so-far unseen data from the test set. These are the final results.

Following these steps prevents *overfitting*, which happens when the model shows good performance on the training data but poor generalization with new data. Also, not using the test set until the end prevents *data snooping bias*. Data snooping happens when the researcher, inadvertently, is biased to choose a particular model or hyperparameter after having a look at the data.

The dataset can be partitioned in different ways. In this work we usually employ, when possible, the 70/20/10 split which means that 70% of the data goes to the training set, 20% to the validation set and 10% to the test set.

2.3.2 Feature Scaling

Machine learning models do not perform well when the input numerical variables have different scales. It is common to perform operations that transform the input and obtain features with similar scales. This is known as *feature scaling*. There are two common techniques [76]:

1. Normalization: The data is rescaled to the range [0, 1]. Given a variable $\mathbf{x}$, a normalized variable $\mathbf{x}'$ can be obtained by

$$\mathbf{x}' = \frac{\mathbf{x} - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})} \tag{2.12}$$

2. Standardization: The mean is subtracted from each feature and then divided by the standard deviation. This makes the values have zero mean and unit variance.

$$\mathbf{x}' = \frac{\mathbf{x} - \bar{\mathbf{x}}}{\sigma} \tag{2.13}$$

where σ is the standard deviation of $\mathbf{x}$

To prevent data snooping, feature scaling should be done for all partitioned datasets but using the minimum and maximum (or mean and variance in the case of standardization) of the training set, since in theory only the training set is known at training time.

2.3.3 Optimization Algorithm

In section 2.2 we mentioned how the weights can be updated by using gradient descent. In practice, it is more convenient to split the dataset in mini-batches (resulting in *mini-batch gradient descent*) or to perform optimization on a single instance at the time (resulting in *stochastic gradient descent*). However, when big datasets are being used, it is preferable to use more advanced optimizers.

In this work we employ for all models the *adaptive moment estimation* optimizer, also known as *ADAM*, which was proposed in [82]. This method requires two vectors, a *momentum vector* $\mathbf{m}$ that keeps track of previous gradients, and a vector $\mathbf{s}$ that accumulates the squares of the gradient $\nabla_\Theta J(\Theta)$. The update process is as follows [76].

1. $\mathbf{m} \leftarrow \beta_1 \mathbf{m} - (1 - \beta_1) \nabla_\Theta J(\Theta)$
2. $\mathbf{s} \leftarrow \beta_2 \mathbf{s} + (1 - \beta_2) \nabla_\Theta J(\Theta) \otimes \nabla_\Theta J(\Theta)$
3. $\hat{\mathbf{m}} \leftarrow \frac{\mathbf{m}}{1-\beta_1^t}$
4. $\hat{\mathbf{s}} \leftarrow \frac{\mathbf{s}}{1-\beta_2^t}$
5. $\Theta \leftarrow \Theta + \alpha \hat{\mathbf{m}} \oslash \sqrt{\hat{\mathbf{s}} + \epsilon}$

Here $\otimes$ represents element-wise multiplication and $\oslash$ represents element-wise division. The hyperparameters β_1, β_2 and ϵ are referred as *momentum decay*, *scaling decay* and *smoothing term* respectively. t is the iteration number.

The ADAM algorithm is much faster than gradient descent, especially in regions where the loss function is flat (plateaus). The inclusion of a momentum vector (which appears originally in [83]) makes the optimization behave as a ball rolling down a slope, meaning that the descent steps will become larger (accelerate) towards the minimum. The hyperparameter β_1 acts as a friction mechanism in order to prevent momentum from growing excessively. On the other hand, the vector $\mathbf{s}$ performs *root mean square propagation (RMSProp)*, which provides an adaptive learning rate that directs the updates towards the optimum.

In this work we use the hyperparameter values recommended in [82], which are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$.

2.3.4 Validation of models

As mentioned in section 2.3.1, the validation step allows us to compare between different trained models. This section discusses a two more advanced validation techniques that are used throughout this theses.

K-folds Cross-Validation

The process previously described uses the validation set for hyperparameter tuning, however using a single validation set for tuning increases the risk of optimizing for that set in particular. For this reason we employ K-folds cross-validation technique. The training and validation sets are combined into one, and split into K roughly equal-sized parts. Then, starting at $k = 1$ the network is fitted by taking the other $K - 1$ folds as the training set and the fold k as a validation set. The resulting model is evaluated and the process is repeated for $k = 1, 2, \cdots, K$ [84]. Figure 2.5 shows a schematic of the process.

Figure 2.5: An schematic of K-folds cross-validation for $K = 5$ and training stage $k = 3$ (adapted from [84]).

After training the model K times the final estimation of the error is done by combining the K estimates. This value is a more accurate estimate of the generalization error when compared to using a single fold [76, 84]. In this work we employ this validation method with $K = 10$.

2.4 Software

The main programming language used in this thesis is Python version 3 [85]. Python 3 supports major scientific libraries, and it is nowadays one of the most

popular languages for data science. Data science and machine learning specific libraries allow many of the algorithms in this section to be used easily without actually implementing the algorithms. Some of these libraries are:

1. Scikit-Learn [86]: Contains many tools and functions for statistical learning and data handling. Feature scaling and dataset partitioning (including K-folds cross-validation splitting) is performed with this library.

2. Scipy [87]: Provides, among other tools, functions for statistical tests. The Wilcoxon signed-rank test is performed with the Scipy library.

3. Numpy [88] and Pandas [89]: provide data structures for efficient data manipulation. Data before and after training is usually stored in either a Numpy array, or a Pandas series or dataframe structure.

4. Matplotlib [90]: Contains tools for the elaboration of plots and graphs. Most of the graphs displaying results in this work were prepared with this library.

The machine learning platform used in this thesis is Tensorflow [91]. Tensorflow was created by Google and is widely used in industry and academia as it provides many tools for research and implementation. It also provides several layers of abstraction that allow us to create models easily by using high-level code, while at the same time design intricate architectures or modified optimization code by using the lower-level code. In this work we use Tensorflow for creating the neural networks, fitting (including backpropagation and optimization) and post-training evaluation.

2.5 Summary of this chapter

In this chapter we introduced the machine learning models and practices uses throughout this thesis. The principal variety to be used is neural networks, and it serves as basis for all of the developed models. The fundamental block of a neural network is the artificial neuron (node) and a network is constructed by stacking nodes into layers.

In its most basic form neural networks are used for function approximation. They are trained by using backpropagation and an optimizer that seeks to minimize a loss function. Common practices for training networks such as dataset partitioning and feature scaling were also introduced.

Neural network in different configurations yield different models targeted at different applications. Two of these models are long short-term memory networks, and generative models. The former is targeted at sequence prediction such as time series, the latter is targeted at probabilistic modeling. Both of this approaches are used throughout this work.

Chapter 3

Methodologies for the NWP and power models

This chapter describes the origin and processing of the measured wind speed data, as well as the numerical weather prediction (NWP) model used to obtain wind speed forecasts. This data will later be used as inputs and targets to our machine learning models. Most of the implementation and development in this chapter is due to Che [12].

3.1 Description of the wind farms of interest

This work uses data from two wind farms located in different locations. The first wind farm of interest is located in Awaji Island, an island in Hyogo Prefecture between the islands of Honshu and Shikoku. Fifteen General Electric GE2.5 turbines are installed in the southern part of the island. The turbines have a rated capacity of 2.5 MW, a rotor diameter of 84 m and a hub-height of 80 m.

Figure 3.1 shows the elevation map of the wind farm area and the location of all turbines. It can be seen that the wind farm is located on rough terrain and the turbines are placed at different elevations ranging from about 80 meters to up to 240 meters above sea level.

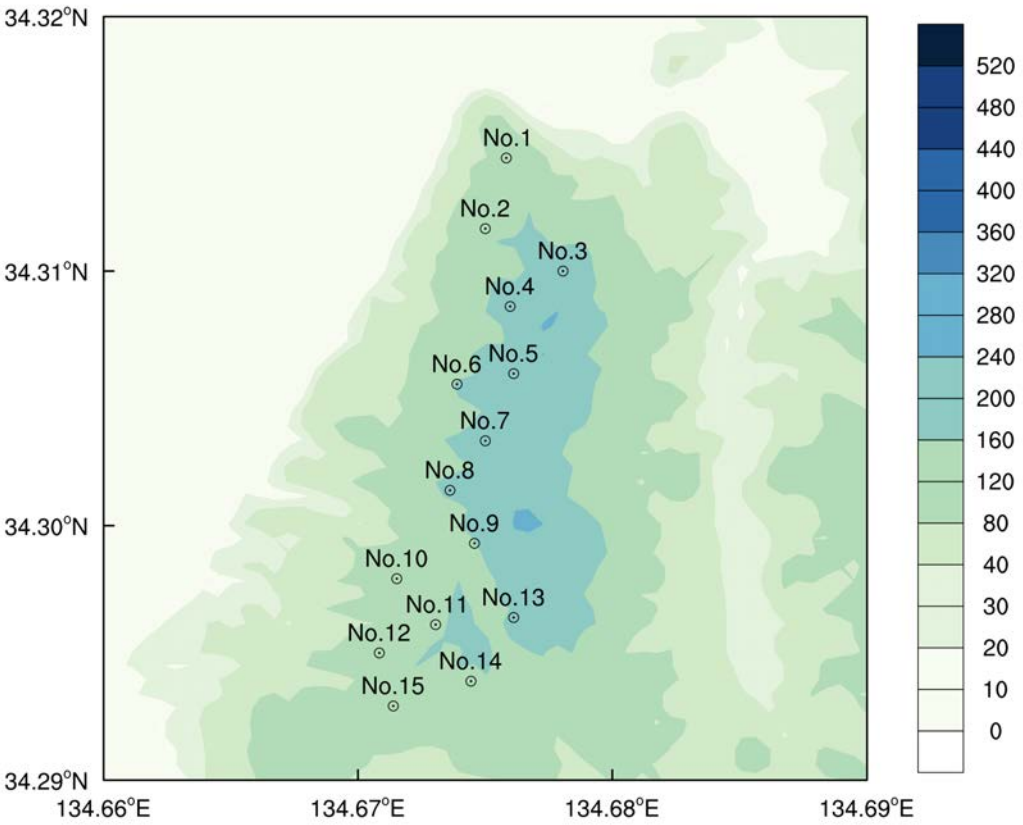

Figure 3.1: Elevation map of Awaji Island (in meters) and location of all 15 turbines.

The second wind farm of interest is located in the Houhoku area in Yamaguchi Prefecture, western Japan. As shown in figure 3.2, 12 turbines are installed in rough terrain with elevations ranging between 120 to 240 meters.

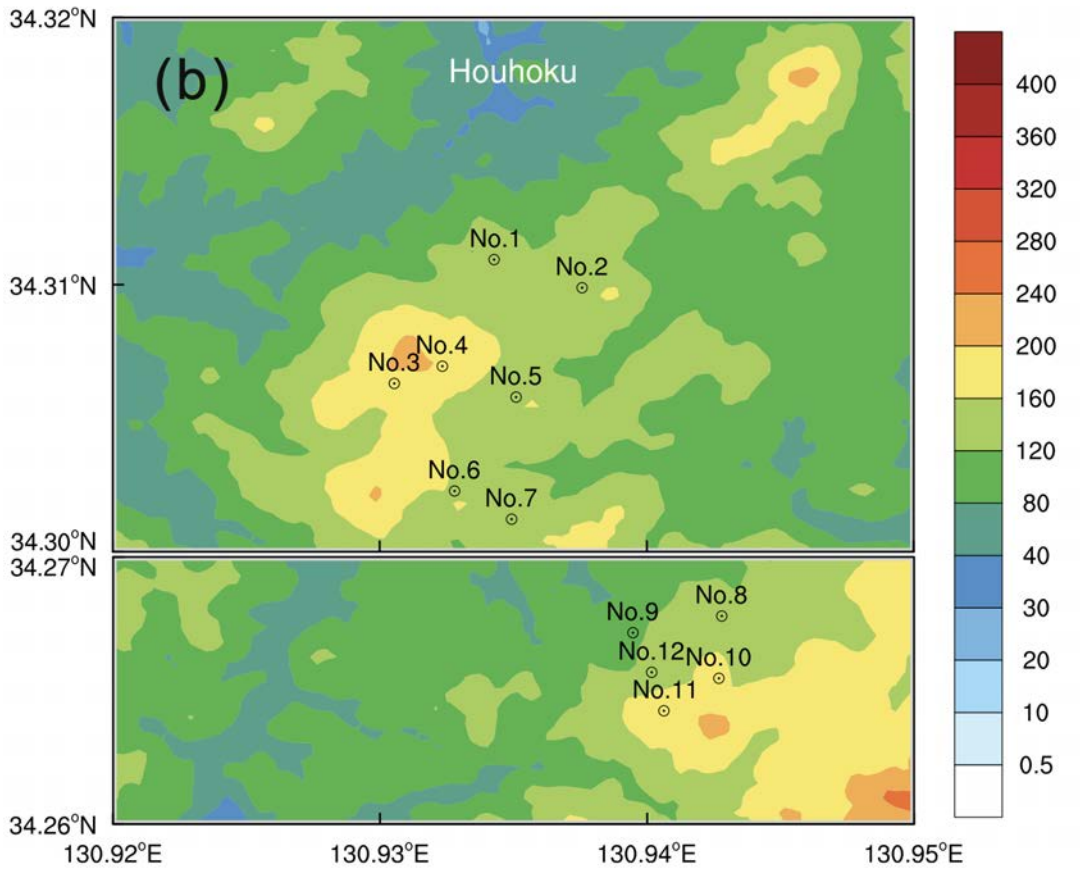

Figure 3.2: Elevation map of the Houhoku Area (in meters) and location of all 12 turbines.

3.1.1 Measurements datasets

Anemometers placed on top of the nacelle behind the rotor of each turbine take measurements of the wind speed after it passes through the blades of the turbines. These measurements are taken as our ground truth in this research and we refer to them as the *observations* or targets. Hourly observations are available for both wind farms, 4.5 years (January 3 2016 to June 1 2020) for the Awaji Island farm, and 5 years (January 1 2015 to December 31 2019) for the Houhoku farm.

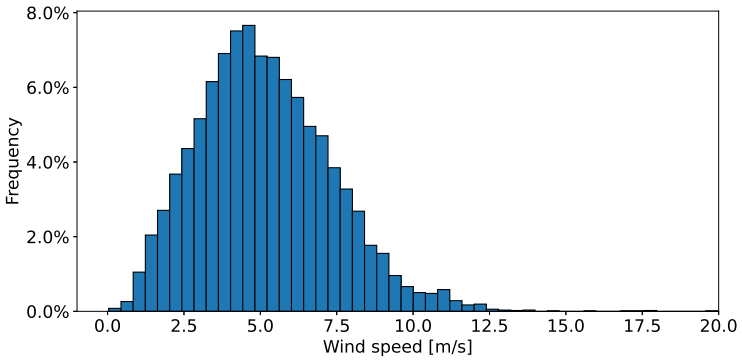

Figure 3.3: Histogram displaying the distribution of wind speed in the Awaji Island farm (aggregated for all turbines).

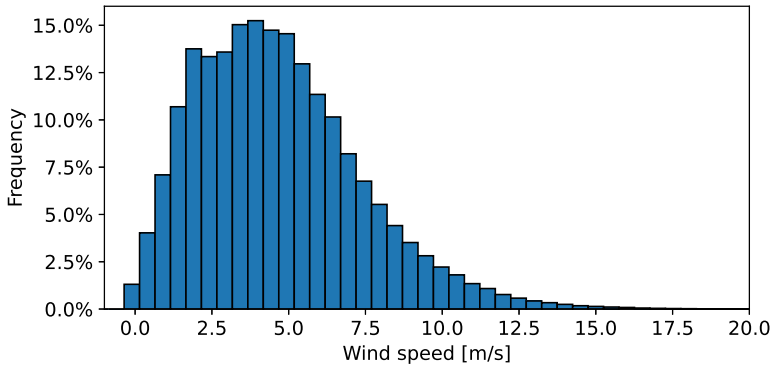

Figure 3.4: Histogram displaying the distribution of wind speed in the Houhoku farm (aggregated for all turbines).

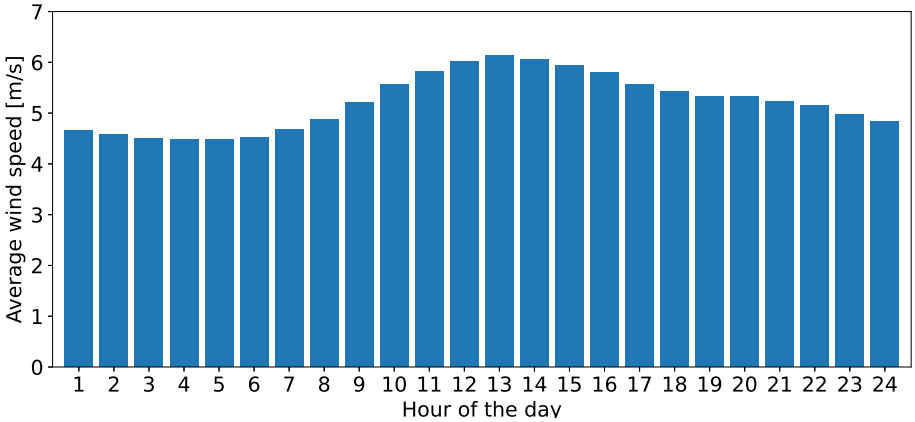

Figure 3.5: Average wind speed per hour of the day in the Awaji Island farm (aggregated for all turbines).

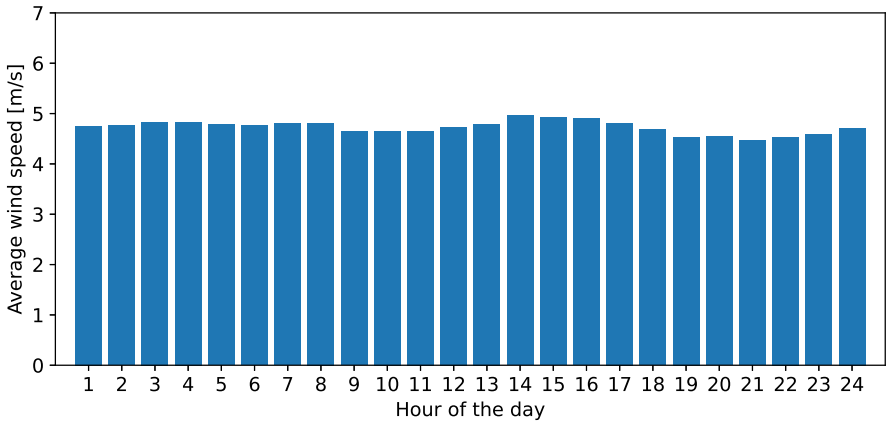

Figure 3.6: Average wind speed per hour of the day in the Houhoku farm (aggregated for all turbines).

Figures 3.3 and 3.4 show the wind speed distribution for both sites, aggregated for all turbines and all measured times. In both cases wind speed approximately follows a Rayleigh distribution with a mean of approximately 4.8 m/s for the Awaji farm and 4.6 m/s for the Houhoku farm. The tails of both distribution extend up to 20 m/s and it is particularly thick between 10 and 13 m/s, which means that strong winds are not uncommon. The same can be seen in figures 3.7 and 3.8 , where boxplots are displayed for each turbine.

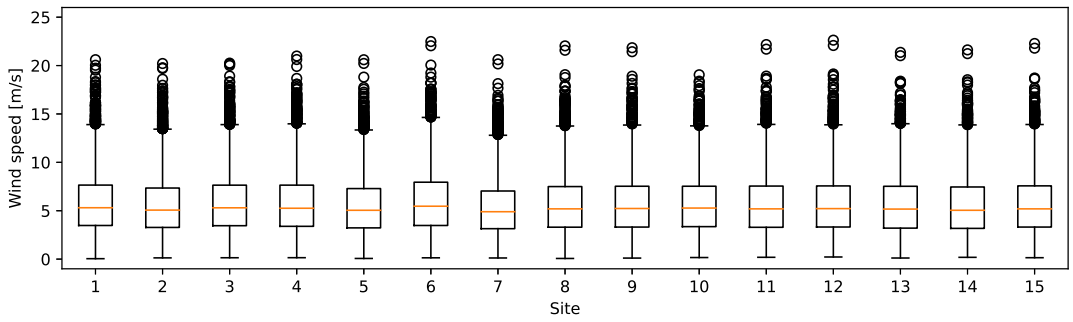

Figure 3.7: Boxplots of wind speed values per site for the Awaji island farm. The upper and lower whiskers are located at 1.5 times the interquartile range (IQR) above the upper quartile (Q3) and below the lower quartile (Q1), respectively. The circles represent the outliers: the values that are greater than Q3+1.5·IQR.

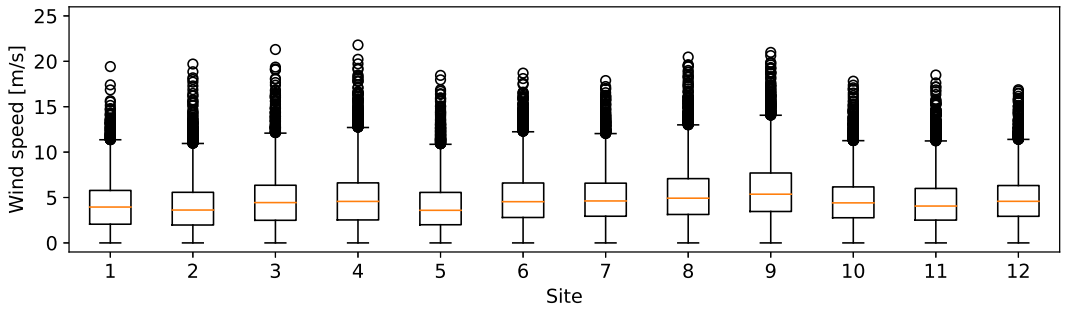

Figure 3.8: Boxplots of wind speed values per site for the Houhoku farm. The upper and lower whiskers are located at 1.5 times the interquartile range (IQR) above the upper quartile (Q3) and below the lower quartile (Q1), respectively. The circles represent the outliers: the values that are greater than Q3+1.5·IQR.

Similarly, figures 3.5 and 3.6 shows the hourly mean wind speed. In the case of the Awaji farm, it can be seen that the strongest speeds occur at 13:00 hours while the weakest winds occur just before dawn, at 5:00. This pattern can also be seen in the boxplot shown in figure 3.9. On the other hand, the Houhoku wind farm shows a more uniform wind speed pattern, with a slight peak at the 14th hour.

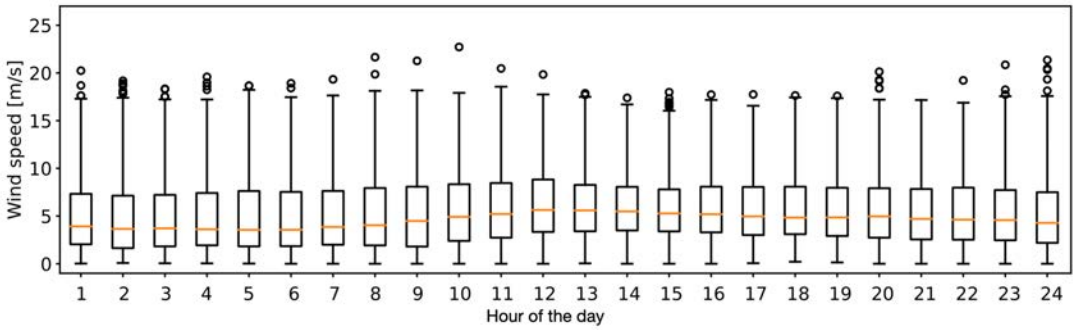

Figure 3.9: Boxplots showing the distribution of the observed data per hour, for the Awaji island farm. The upper and lower whiskers are located at 1.5 times the interquartile range (IQR) above the upper quartile (Q3) and below the lower quartile (Q1), respectively. The circles represent the outliers: the values that are greater than Q3+1.5·IQR.

Measuring wind speed after it passes the nacelle and rotating blades introduces uncertainty, therefore, nacelle-based measurements are less reliable for wind farm

turbine control. Ideally, measurements from nearby upwind meteorological towers should be employed [12]. Nacelle-based measurements can be adjusted to include the effects of the nacelle and rotor blades by assimilating data from nearby meteorological towers [92], however no meteorological data from nearby towers was readily available for this study. Moreover, according to Cutler et al. [93], nacelle-based measurements might me more reliable than tower measurements located at far distances.

Finally, the measured data needs to go under a quality control process in order to flag inconsistent values or outliers. These bad data points can degrade the skill of the postprocessing models. Therefore, the measurements are evaluated with the standards in the technical report of NOAA Earth Systems Research Laboratory [94].

3.2 The WRF model

The Weather Research and Forecasting (WRF) model is a mesoscale numerical weather prediction model designed for both atmospheric research and operational forecasting [95]. The WRF system itself consists of three major systems:

1. The WRF Preprocessing System (WPS): This system preprocesses all necessary inputs to run the WRF model properly. At this stage is also where simulation domain configuration, degribbing, topography data interpolation and meteorological data interpolation have to be specified.

2. The Core WRF model: Within the core WRF model there are two dynamical solvers: the Advanced Research WRF (ARW) solver and the Nonhydrostatic Mesoscale Model (NMM) solver.

3. The post-processing and visualization system: The output from the core model comes in format NETCDF. The postprocessing and visualization system provides packages to create outputs for simple visualization such as GrADS or Vis5D.

3.2.1 Implementation in Awaji Island

The WRF model with the ARW core was implemented in the zone of the Awaji Island wind farm. The domain configuration consisted of one parent domain (D01) and three nested domains (D02, D03 and D04) as shown in figure 3.10. Table 3.1 shows the resolution of each domain and the dimensions in grid points. The vertical resolution for all domains is 35 vertically stretched eta levels, 10 of which are within the lowest 1 km.

The topography and elevation data was obtained from the U.S. Geological Survey datasets, with a resolution of 30 arc seconds for all domains (roughly 762m x 762m, calculated for a latitude of 34.65^{o} N).

Among the physical options chosen to run the model are the WRF Single-Moment 6-class microphysics parameterization scheme [96]; the new Kain-Fritsch convective parameterization [97]; the Noah land surface model [98]; Dudhia short-wave radiation [99] and Rapid Radiative Transfer Model longwave radiation [100]; and the Asymmetric Convective Model version 2 planetary boundary layer scheme [101].

Figure 3.10: Map showing the parent and nested domains for the WRF implementation. The red triangle in the innermost domain shows the location of the wind farm of interest.

Table 3.1: Mesh resolution for each of the domains

Domain	**Mesh Resolution**	**Mesh Grid Points (lon x lat)**
D01	13.5 km	146x104
D02	4.5 km	193x124
D03	1.5 km	175x106
D04	0.5 km	172x82

The model was run for the same period for which observations are available, that is from January 3 2016 to June 1 2020. For every simulation day, the model was initialized in "cold-start" at 18:00 UTC (03:00 JST) and each simulation ran for 30 hours. Outputs were extracted at 1 hour intervals, with the first 6 hours (spin-up time) discarded, as recommended in [95]. The final extracted forecast consists of the 24 h between 00:00 UTC (09:00 JST) and 23:00 UTC (08:00 JST).

3.2.2 Initial and boundary conditions

Initial and boundary conditions are obtained from the Global Forecast System (GFS), a NWP model that covers the entire globe and provides deterministic and probabilistic forecasts for up to 16 days. It is produced and maintained by the National Centers for Environmental Prediction. Forecasts are available four times a day at six hour intervals. Initial and boundary data for was taken from the GFS at a horizontal resolution of 0.5 x 0.5 degrees and a vertical resolution of 32 levels, from 1000hPa to 1hPa for all variables.

3.3 Wind power forecasts

In section 1.3.4 we provided a brief review of the common techniques for wind power estimation. In this section we present the concrete methods employed in this research for power data preprocessing and power estimation.

3.3.1 Data preprocessing

Regardless of the chosen model, measured data has to be assessed before its used. There are instances where the power output is negative which implies the wind turbine is consuming energy due to low wind speed, or cases where the power output is variable even if wind speed is constant [30]. These occurrences can cause models to fail, and therefore any outliers must be removed before model training.

In this work we employ Tukey's method [102] for the removal of these outliers. This method, also known as the 1.5xIQR rule, uses the quartiles of the data to find values that lie outside the overall pattern in the data. The process is as follows:

1. Find the quartiles of the data: Q_1, Q_2, Q_3 and Q_4.
2. Find the interquartile range: $\text{IQR} = Q_3 - Q_1$
3. For ever x in the dataset, apply the rule: if $x < Q_1 - 1.5\text{IQR}$ or $x > Q_3 + 1.5\text{IQR}$, then x is an outlier.

For the removal of outliers in power data we apply Tukey's method both in the power and speed axis. Each axis was partitioned into bins and the outliers in each bin were removed taking into account the quartiles of each bin. Figure 3.11 shows the results for one of the turbines, where the red points are the outliers identified with Tukey's method and the bin approach. It should be mentioned that as a final step, some outliers had to be removed manually.

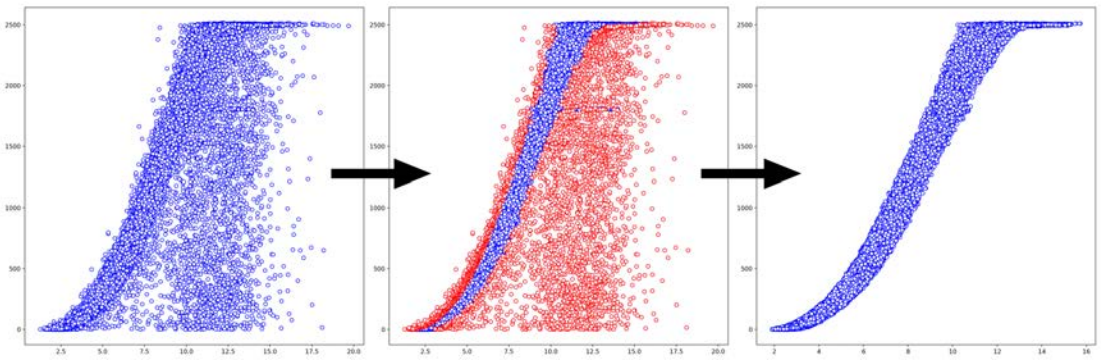

Figure 3.11: Example of outlier removal with Tukey's method, showing the data of one of the turbines in the wind farm of interest.

3.3.2 Power-curve model

After removing the outliers we employed a non-linear regression model to fit the power and speed data to a function. The chosen model was proposed by [103] and belongs to the parametric model family as described in section 1.3.4. The expression is shown in equation 3.1.

$$p(u) = p_{\max}\left(1 + \left(\frac{\beta}{u}\right)^{\alpha}\right)^{-k} \tag{3.1}$$

Here $p_{\max}$ is the maximum power of the turbine and α, β and k are the parameters to be estimated. Although $p_{\max}$ is a known variable, in this research it is also included as a parameter to optimize. Figure 3.12 shows the result of this model fitted in the example data from the previous section. It can be seen that the approximation (black line) fits the data well, and is mostly centered around the points of real measured data.

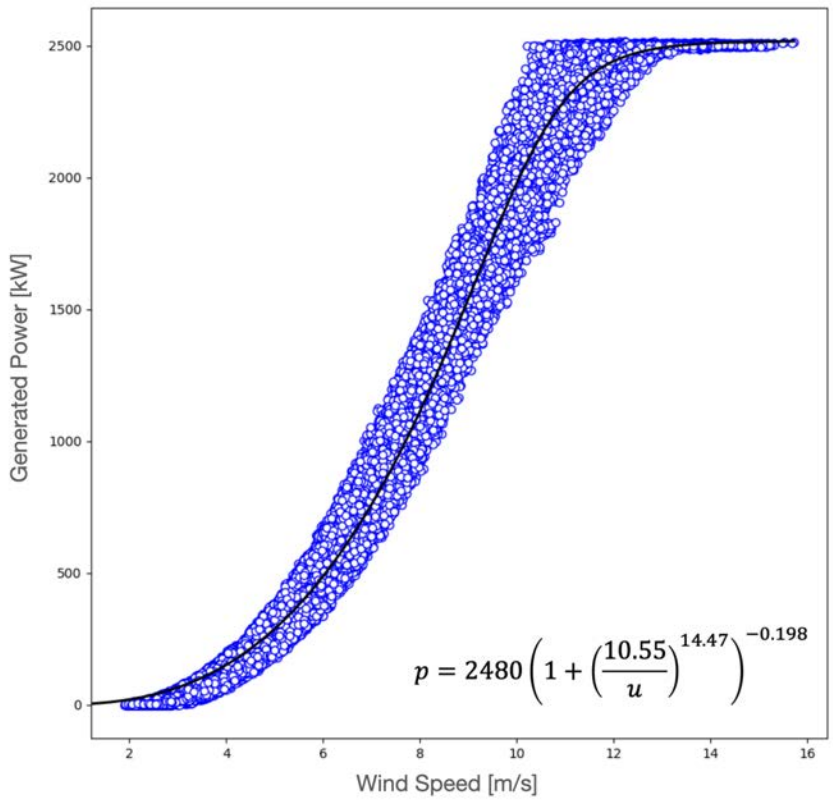

Figure 3.12: Curve fitting using equation 3.1 on the data shown in figure 3.11. For this case the estimated parameters converged at $p_{\max} = 2480$, $\alpha = 14.47$, $\beta = 10.55$ and $k = 0.198$.

3.4 Evaluation tools

This section describes the metrics, methods and tools that are used to evaluate and assess the models. These are presented first according to their nature (deterministic or probabilistic) and second according to their kind (scalar metric or graphical tool). In this section we denote any real data with y and any estimation (from the NWP model or postprocessing model) with $\hat{y}$.

3.4.1 For deterministic models

Scalar metrics

The main metric used to evaluate deterministic models will be the root mean square error (RMSE), the mean absolute error (MAE) and the correlation coefficient:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(y_k - \hat{y}_k\right)^2} \tag{3.2}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{k=1}^{N}\left|y_k - \hat{y}_k\right| \tag{3.3}$$

$$\mathrm{CC} = \frac{\sum_{k=1}^{N}\left(y_k - \overline{y}\right)\left(\hat{y}_k - \overline{\hat{y}}\right)}{\sqrt{\left(\sum_{i=1}^{N}\left(y_k - \overline{y}\right)^2 \sum_{i=1}^{N}\left(\hat{y}_k - \overline{\hat{y}}\right)^2\right)}} \tag{3.4}$$

where N is the number of samples and the bar denotes the mean of the corresponding variable. To find the improvement over the NWP forecast after applying a postprocessing method, we use the percentage of change (called *improvement* throughout this work) and also the the skill score (SS) as defined by Wilks [19]:

$$\mathrm{SS} = \frac{A - A_{\mathrm{ref}}}{A_{\mathrm{perf}} - A_{\mathrm{ref}}} \times 100\% \tag{3.5}$$

where A metric after postprocessing, A_{ref} is the metric before preprocessing (the reference metric obtained with the NWP data) and A_{perf} is the metric of a perfect forecast. For both RMSE and MAE, $A_{\mathrm{perf}} = 0$.

Graphical tools

The Taylor diagram [104] is a graphical tool that allows to visualize the discrepancy in centered RMSE, correlation and standard deviation between two forecasts and a reference, by using the geometric relationships between these three metrics. Figure 3.13 shows an example for three datasets: two forecasts, A and B, are being compared to a reference "Obs." Each dataset is represented by a marker in the diagram. The angular coordinate shows the correlation between the forecast and the reference, which ideally should be equal to one. The radial coordinate shows the standard deviation of the forecast, which ideally should be equal to the standard deviation of the reference (shown by the blue broken-line arc centered in 0 and with radius "Obs."). Finally, the linear distance between the "Obs." and the marker for each model corresponds to the centered RMSE (shown with the concentric arcs centered in "Obs."). It can be seen therefore that forecast can easily be judged by their position in the diagram; generally the closer a forecast marker is to the reference marker, the better the metrics for that forecast are.

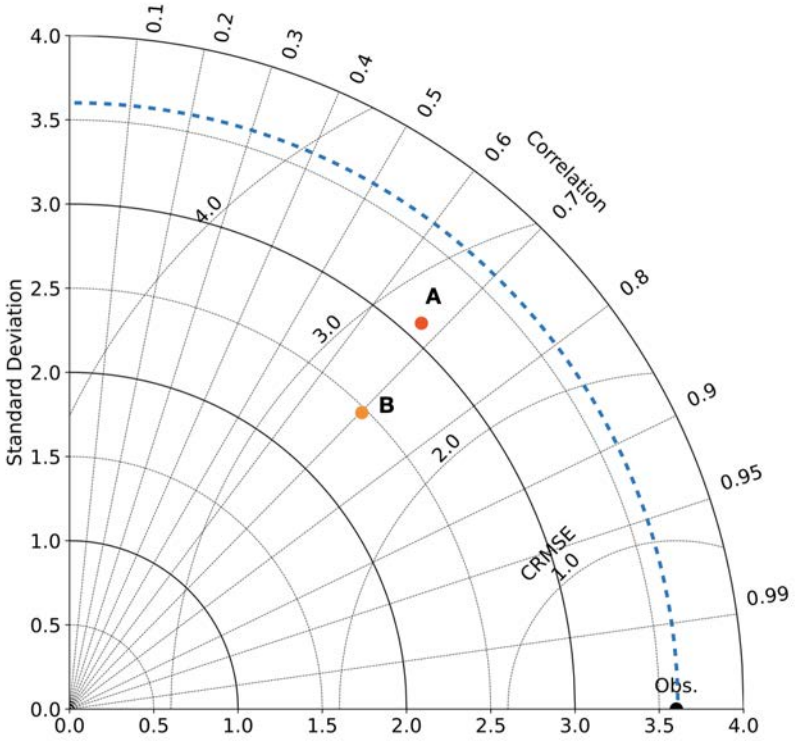

Figure 3.13: Taylor diagram example. Forecast B shows slightly better correlation than forecast A, although A's standard deviation is closer to that of the reference. Forecast B shows lower centered RMSE than A.

3.4.2 For probabilistic models

Scalar metrics

The first metric we use to assess the probabilistic models is the continuous ranked probability score (CRPS) [105]. Supposing that the variable of interest it denoted by y and that the cumulative distribution function (CDF) of the probabilistic forecast is given by F, the CRPS is given by

$$\mathrm{CRPS} = \int_{-\infty}^{\infty} \left[F\left(y\right) - F_o\left(y\right)\right] dz \tag{3.6}$$

where F_o is a cumulative-probability step function that changes from 0 to 1 when y equals the observed value, i.e.

$$F_o(y) = \begin{cases} 0 & \text{if} \quad y < \text{observed value} \\ 1 & \text{if} \quad y \geq \text{observed value} \end{cases} \tag{3.7}$$

It is difficult to directly evaluate Eq. 3.6, especially in the context of ensemble forecasts. Therefore, in the present thesis the CRPS is calculated using the decomposition method presented in [106]. The CRPS value is negative-oriented (the smaller the values, the better) [19], and it rewards ensembles which CDFs match the step function centered at the observation, as shown in figure 3.14.

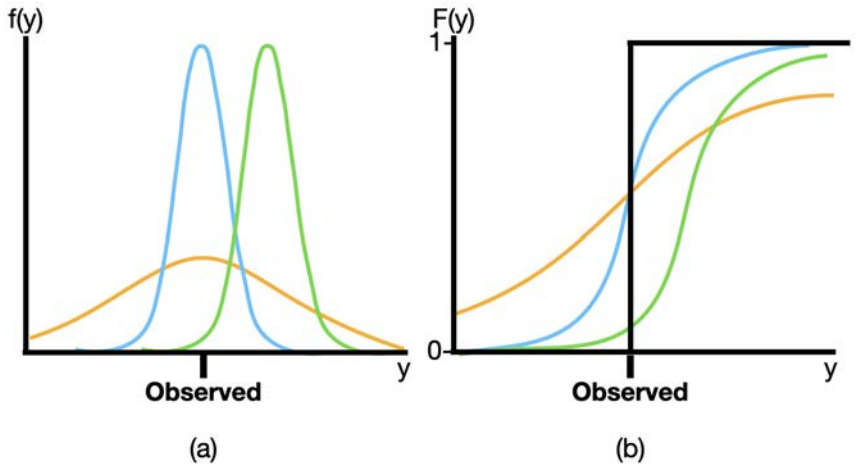

Figure 3.14: Schematic of the CRPS given three probabilistic forecasts and one observation (adapted from [19]). Panel (a) on the left shows the three different PDFs while panel (b) on the right shows their corresponding CDFs. The blue distribution has the smallest CRPS since its PDF is highly concentrated around the observation, and this means that the difference between its CDF and the step-function CDF is the smallest. On the other hand, the green and yellow distributions have larger CRPS since due to not being centered around the observation (as in the case of the green distribution) or due to overdispersion (as in the case of the yellow distribution).

Another characteristic of a good ensemble forecast is *uniformity*, i.e. the observation should behave like another random draw from the same distribution that produced the ensemble [107]. One way to quantify the uniformity of an ensemble is by using the reliability index (RI), given by the following equation.

$$\mathrm{RI} = \sum_{j=1}^{n_{ens}+1} \left| f_j - \frac{1}{n_{ens}+1} \right|, \tag{3.8}$$

where f_j is the observed relative frequency of rank j and $n_{\mathrm{ens}} + 1$ is the size of the ensemble [108, 109, 110]. Similarly to the CRPS, the reliability index is

negative-oriented. The RI score is related to the rank histogram plot, which will be introduced in the next section.

Since both the NWP model and the postprocessing methods will produce day-ahead 24 hours forecasts, it is of interest to also carry on evaluations for each forecasted day as a whole. Therefore, the next two probabilistic metrics to be used are multi-variable, meaning that each daily forecast is considered a vector quantity and each hour is a different variable within that vector.

First, the energy score (ES) is employed. Introduced by Gneiting and Raftery [111], it was designed as a generalization of the CRPS for vector valued quantities. Here the energy is calculated as in Gneiting et at. [110], where the score for a particular day d is given by

$$\mathrm{ES}_d = \frac{1}{n_{ens}} \sum_{i=1}^{n_{ens}} ||\hat{y}_d^i - y_d|| - \frac{1}{2n_{ens}^2} \sum_{i,j=1}^{n_{ens}} ||\hat{y}_d^i - \hat{y}_d^j||, \tag{3.9}$$

where $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{n_{\mathrm{ens}}}$ are the members of an ensemble forecast. Second, the variogram score (VS) is also used as proposed by Scheuerer and Hamill [112]. The VS for a given day d is:

$$\mathrm{VS}_d = \sum_{k,k'}^{T} w_{kk'} \left(|y_{d,k} - y_{d,k'}|^\gamma - \frac{1}{M} \sum_{i=1}^{M} \left| \hat{y}_{d,k}^i - \hat{y}_{d,k'}^i \right|^\gamma \right)^2, \tag{3.10}$$

where k and k' correspond to each forecast lead time and therefore $T = 24$ and $w_{kk'}$ are non-negative weights. For all lead times the weights $w_{kk'}$ are set to 1 and the parameter γ is set to 0.5 as recommended by Scheuerer and Hamill [112] and Dumas et al. [73].

Both of these multivariate scores quantify the skill of the models to correctly capture the means and variances of the assumed distributions. The variogram score also provides information about the correlations between the forecast lead times [73]. Both are negative oriented.

Graphical metrics

A graphical method to test for statistical consistency and uniformity in an ensemble is to employ rank histograms [113]. A rank histogram is plotted by concatenating all members of a ensemble and the observation, ranking all values, and then finding the rank of the observation [114, 115, 113]. In a statistically consistent

ensemble, the rank of the observation is equally likely to be any integer between 1 and $n_{\text{ens}}+1$, meaning that the observation should be statistically indistinguishable from the ensemble members.

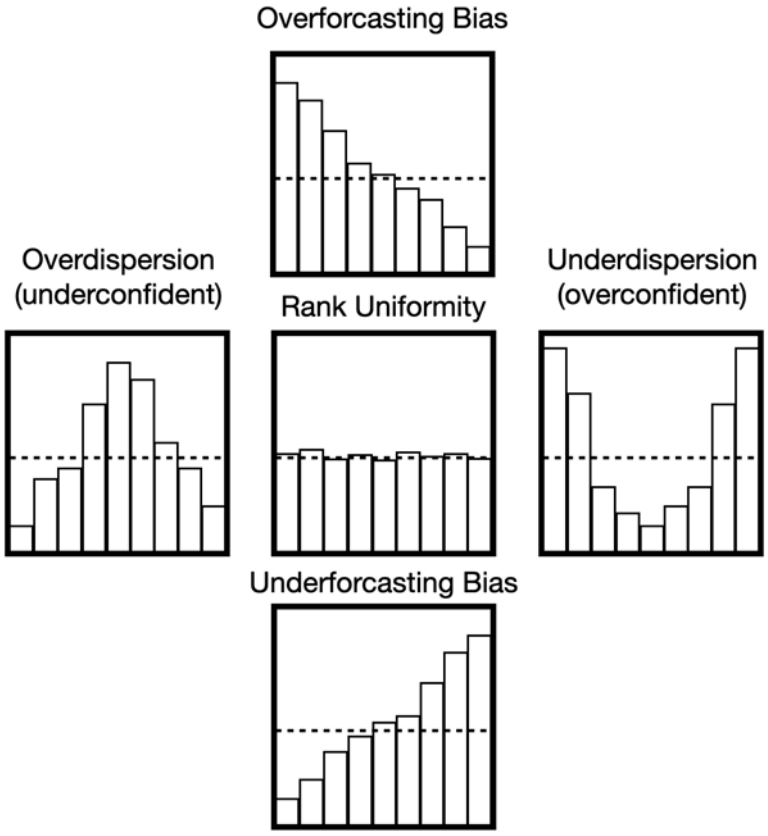

Figure 3.15: Examples of rank histograms and their interpretation. Adapted from [19].

Figure 3.15 shows several examples of rank histograms for an ensemble of size $n_{ens} = 8$. The horizontal dashed line shows the ideal case in which the ensemble shows perfect uniformity. Ensembles with underforecasting or overforecasting biases usually show overpopulation of the extreme ranks in the right or left respectively. Overdispersed ensembles show overpopulation in the middle ranks, while underdispersed ensembles show overpopulation in both extremes. As mentioned before, the RI score is directly related to the rank histogram; the lower the RI the more uniform the rank histogram is.

Finally, it is also necessary to study the dispersion characteristics of the ensembles [113, 116]. Let $\hat{y_1}, \hat{y_2}, \ldots, \hat{y}_{n_{\text{ens}}}$ be an ensemble forecast for time t, $\bar{\hat{y}_t}$ be its mean, and s_t^2 be its sample variance, If the ensemble forecast is statistically

consistent, then the following property will be held true:

$$\mathbb{E}_t\left[\left(y_t - \bar{\hat{y}}_t\right)\right] = \frac{n_{\text{ens}} + 1}{n_{\text{ens}}}\mathbb{E}_t\left[s_t^2\right], \tag{3.11}$$

where y_t is the observed value. Eq. 3.11 basically states that in a statistically consistent forecast the mean square error between the observations and the mean ensemble must be equal to the mean of the variances, multiplied by a size-dependent scaling factor [117]. We can test for this relationship by plotting both the MSE and the variance in a dispersion diagram.

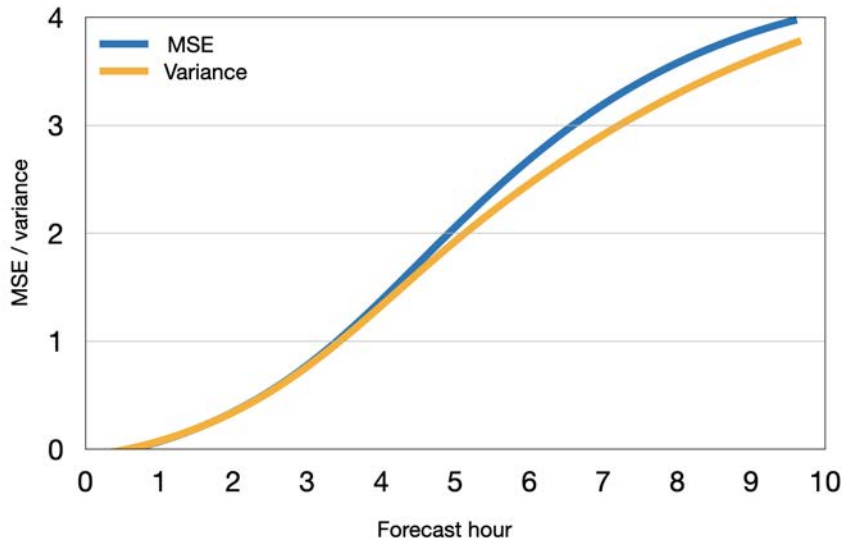

Figure 3.16: Example of a dispersion diagram (adapted from [117]).

Figure 3.16 shows an example of a close-to-ideal dispersion diagram. Normally the metric is calculated for every forecast lead time (days in this example) in order to visualize the evolution of the MSE with respect to simulation time.

3.5 Summary of this chapter

In this chapter we describe the origin of the data (both observations and predictions), power model and evaluation tools used in this work. The wind speed and power data for the training of the correction models comes from a wind farm located in Awaji Island, a zone with rugged terrain and complex landform that results in highly fluctuating wind speed patterns. Measured wind speed data was

obtained for each of the 15 turbines in the wind farm, by using anemometers positioned on top of the nacelle after the rotating blades. Wind speed data was obtained for a period of 4.5 years.

Numerical weather predictions were obtained for the same period of time. The chosen NWP model was the Weather Research and Forecasting (WRF) model, using initial and boundary conditions from the Global Forecast System (GFS) and a 4-domain grid implementation. With respect of the power estimation model, a non-linear power-curve model was chosen to fit wind and power data. The procedure to pre-process the power data was also described.

Finally, we introduced all the evaluation tools to be used in this work. Both deterministic metrics and probabilistic metrics were briefly described, as well as the graphical tools and visualization techniques used to assess the performance of the models after each task.

Chapter 4

An LSTM model with variational mode decomposition

In this chapter we present a data-driven hybrid approach for the prediction of wind speed. The two main components of this approach are a signal decomposition step, followed by a machine learning-based prediction step. The intent in this chapter is to explore pure data-based forecasting by using machine learning methods, and establish their advantages and disadvantages for wind speed forecasting. We start with a description of its nature and training details. Results are shown in section 4.3. A discussion about the results is offered at the end of the chapter.

4.1 Description of the model

The model introduced in the present chapter explores the idea of using signal processing tools in wind speed time series. Unlike the models that will be introduced in the next chapters, the model in this chapter deals with the task of *prediction* rather than *correction*. This means that this model is designed for obtaining 24-hours day ahead wind speed predictions directly from past data without the forecast of the NWP model.

Within the time series forecasting context, treating a time series as a signal and decomposing it into sub-signals is a common approach, sometimes employed as a preprocessing step before the prediction task. The idea is that the resulting sub-signals will present more stable and predictable patterns than the original signal,

specially in the case of stochastic time series such as wind speed. Some decomposition methods that have been used in wind speed forecasting are the empirical mode decomposition [118, 119], principal component analysis [120], wavelet decomposition [121, 122, 119, 123, 124, 125] and singular spectral analysis [126, 119]. Most of these decomposition methods allow recombination of the sub-signals to obtain the original signal. Therefore, by performing prediction on each sub-signal independently and recombining them all we can obtain a prediction for wind speed.

In this chapter we partially follow the approach presented by Rodrigues et al. (2020) [20], were the variational mode decomposition (VMD) method is used a part of a multistage forecasting model. The prediction model for each of the sub signals is long short-term memory (LSTM) networks, already presented in section 2.2.1. A LSTM network is trained separately for each mode. Therefore, we refer to this approach as VMD+LSTM.

The VMD+LSTM prediction task of 24-hour wind speed for day number d is summarized as follows.

1. Collect the observations data from m days before, that it days $d-m$ to $d-1$. Data is stored in an array of size $24 \times m$.

2. Perform VMD decomposition on the $24 \times m$-sized array. Obtain K modes u_k and K central frequencies ω_k.

3. For each mode obtain a 24-hour prediction by using its respective LSTM network.

4. Recombine the predictions.

In this work we choose $m = 5$ and $K = 5$, meaning that after decomposition we obtain 5 arrays of size 120. In the remaining part of this section we briefly introduce the VMD method, and we also explain the architecture of the LSTM networks employed.

4.1.1 Variational mode decomposition

The variational mode decomposition (VMD) method was introduced by Dragomiretskiy and Zosso [127] as a way to mathematically formalize the empirical mode de-

composition (EMD) [128] introduced in the late 1990s. The goal is to decompose a real valued input signal into modes that have specific sparsity properties, and that also allow to reconstruct the input.

Under this context, a mode is defined as a signal whose number of local extrema and zero-crossings differ at most by one [128]. These can be written as amplitude-modulated-frequency-modulated (AM-FM) signals:

$$u_k(t) = A_k(t)\cos(\phi_k(t)) \tag{4.1}$$

where the phase $\phi_k(t)$ has to be non-decreasing, the envelope $A_k(t)$ has to be non-negative, and the envelope $A_k(t)$ and the instantaneous frequency (the derivative of the phase, that is $\phi_k'(t)$) have to vary much slower than the phase itself. Under this conditions it is safe to assume that, on a sufficiently long interval $[t-\delta, t+\delta]$, $\delta \approx 2\pi/\phi_k'(t)$, the mode behaves as pure harmonic with amplitude $A_k(t)$ [128, 129, 130].

This definition implies that each node will have limited bandwidth, that is, each node k is assumed to be mostly compact around a center pulsation ω_k. This frequency can be estimated for each mode u_k by means of the following process.

1. Using the Hilbert transform to calculate the associated analytic signal and obtain a unilateral frequency spectrum.

2. Mix the spectrum with an exponential term to shift it to baseband.

3. Estimate the bandwidth by using Gaussian smoothness, that is, the squared L^2 norm of the gradient.

Finally, to decomposed the signals the following constrained variational problem has to be optimized.

$$\min_{\{u_k\},\{\omega_k\}} \left\{ \sum_k \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\| \right\} \quad \text{subject to} \quad \sum_k u_k = f \tag{4.2}$$

Here, δ represents the Dirac function, $*$ is the convolution operator and $j = \sqrt{-1}$.

After decomposing the original vector of size 120 we obtain K sub-signals of the same size. The recombination of these modes will yield an approximation to the original signal. An additional vector, the *residue*, can be obtained by subtracting the reconstruction from the original signal. Therefore, we obtain $K + 1$ sub-signals. Figures 4.1, 4.2 and 4.3 show examples of 6-days wind speed time series decomposed into 5 nodes and the residue r.

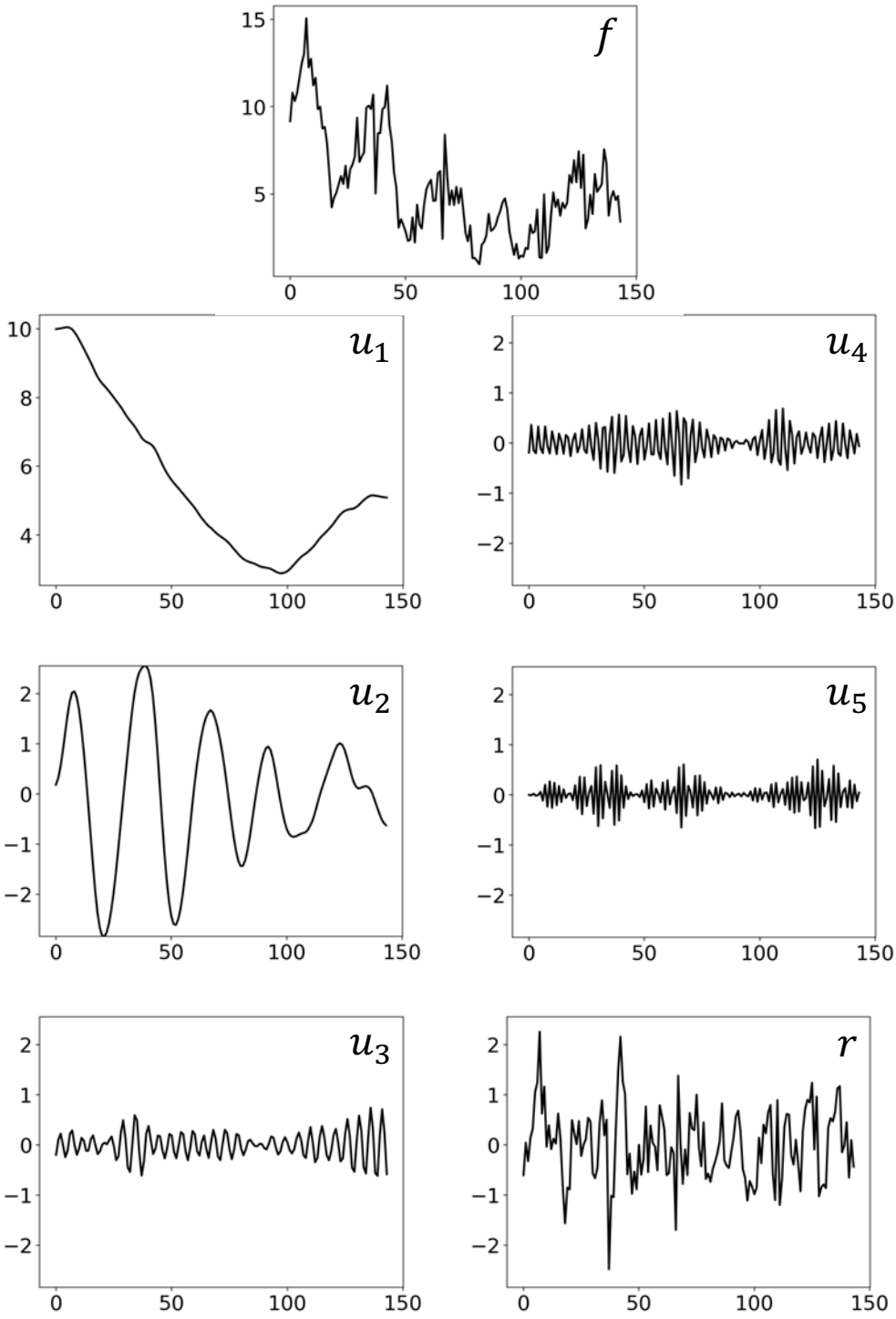

Figure 4.1: Example 1 of VMD decomposition for a 6-day wind speed time series (144 points), into 5 modes plus the residue.

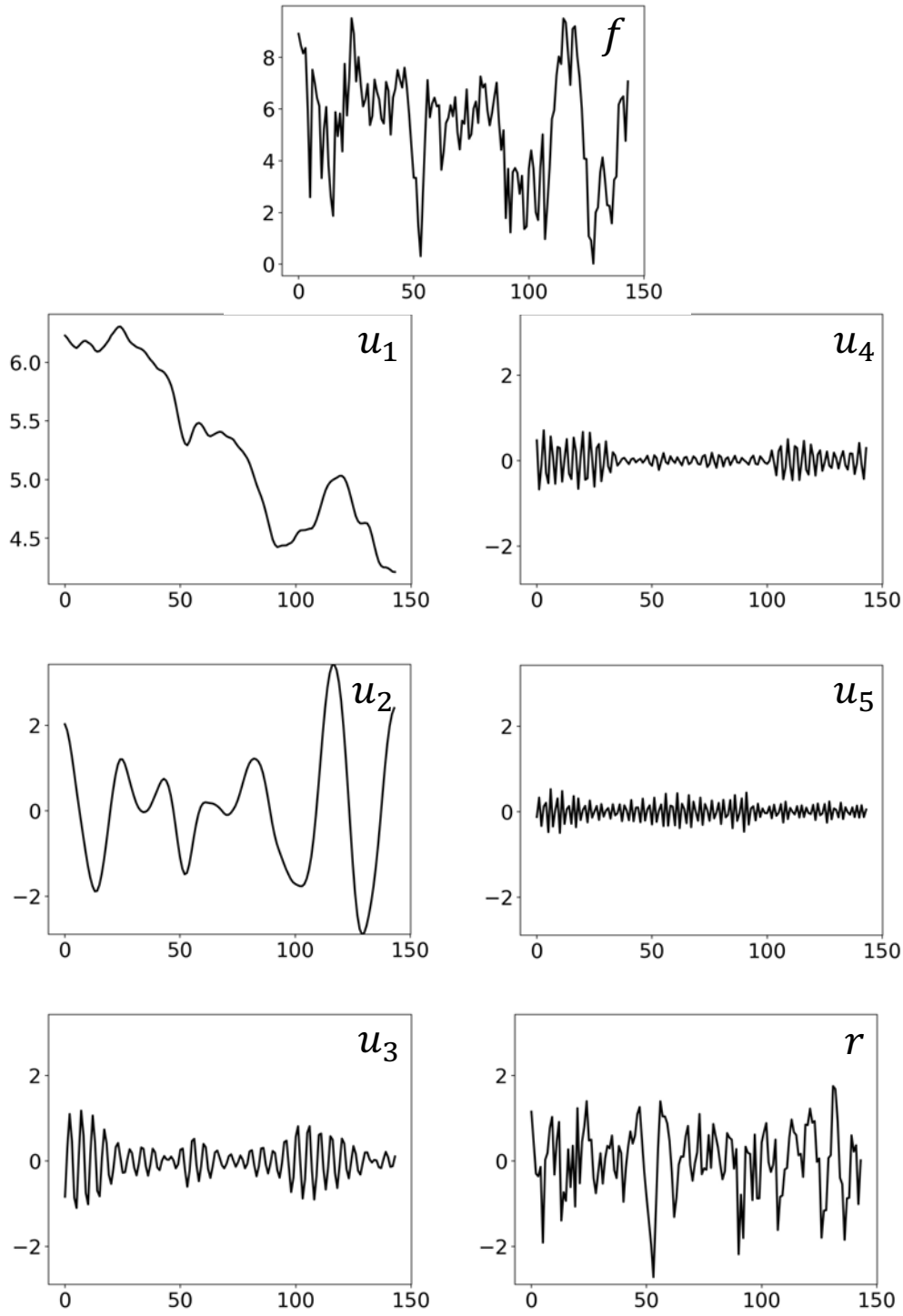

Figure 4.2: Example 2 of VMD decomposition for a 6-day wind speed time series (144 points), into 5 modes plus the residue.

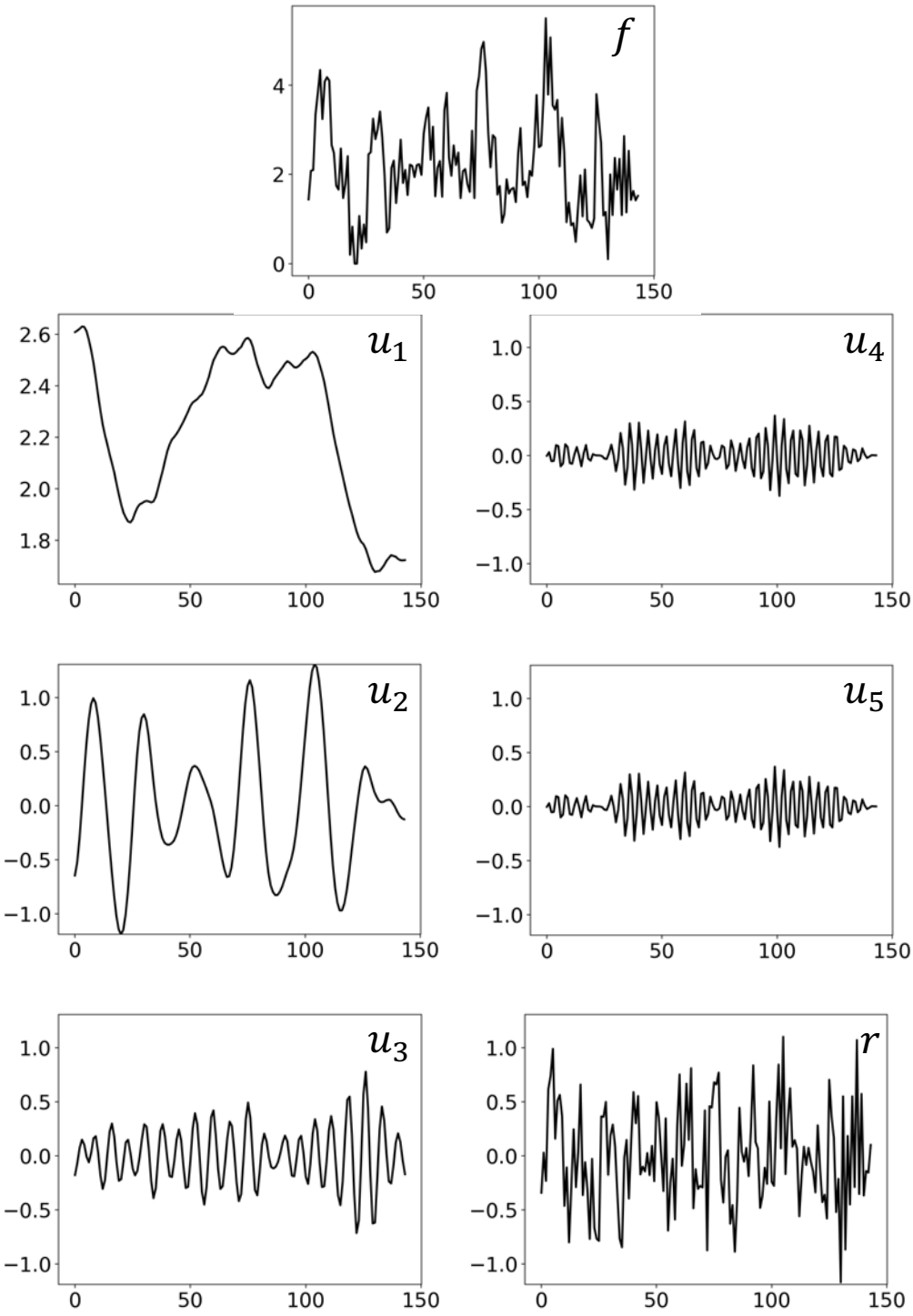

Figure 4.3: Example 3 of VMD decomposition for a 6-day wind speed time series (144 points), into 5 modes plus the residue.

4.1.2 Architecture

A different LSTM network is created for each node and for the residue, but all of them have the same architecture. An input layer of 120 nodes is followed by two hidden LSTM layers of 64 nodes and 128 respectively. The output layer is a fully connected layer of one node. This means that the prediction process is done recursively by concatenating the output of the network to its input and feeding it again as input. This is done 24 times to obtain a 24-hour forecast.

It should be mentioned that the 64-node layer is configured to provide the entire output sequence (the output for every input time step) as input the 128-layer. On the other hand, the 128-layer outputs only the last time step to the fully-connected layer.

4.2 Training details

For this section we employ the observations from the wind farm located in the Houhoku region, as described in chapter 3. This wind farm is located in a zone with rugged terrain and wind in the area shows unstable patterns. The process for data collection and quality control are described in section 3.1.

All of the data is partitioned into 6-day ($m + 1$) segments, corresponding to the day of interest (target day) and the past 5 days (history data). This dataset was then split into two sets for training and testing:

1. Training: 2015-01-01 to 2018-12-31 (aprox. 80% of data)

2. Testing: 2019-01-01 to 2019-12-26 (aprox. 20% of data)

The VMD process involves convolution. Therefore, for any decomposed mode u_k the data at time t will contain information from a time window $t - \Delta t/2 < t < t + \Delta t/2$ from the original signal. The process to construct the features and targets has to take this into account in order to prevent *data leakage*, i.e., to erroneously train the model with information that will not be available at prediction time. If we perform VMD decomposition on the 6-day segments and then split each mode into features (first 120 hours) and targets (last 24 hours), the final hours of the feature vectors will inevitably contain information of the last 24 hours of

the original signal. The resulting model will produce optimistic estimates during training and testing, but it will fail during deployment in the real world.

To avoid this problem we fabricate our features and targets in the following manner. Features are obtained by performing VMD on the history data part of each segment only (first 120 hours). This is used as the input of the model. Targets are obtained by performing VMD on the whole 6-day segment, but only the last 24 hours are used as targets for the model. Concatenating the features and the targets for each mode will result in a discontinuity, however, we consider this a preferable alternative.

For the implementation of the VMD method in our wind speed prediction task we use a Python translation of the original code provided by the developers of the method [127]. In this code the necessary parameters are a balancing parameter α, a noise-tolerance constant τ, the number of modes K, the initialization values for ω_k, a boolean describing if the first mode is put and kept at DC, and a value for tolerance of convergence. In this work we employ $\alpha = 2000$, $\tau = 0$ and $K = 5$, all ω_k start uniformly distributed, the first mode is not kept at DC and tolerance is 10^{-7}.

Regarding the training of the LSTMs, each network was trained for 20 epochs using the ADAM optimizer described in section 2.3.3 and using the MSE as a loss function. The features were scaled by using standardization (section 2.3.2)

4.3 Results

In this section we present the results for the prediction of 24-hour wind speed forecasts by using the VMD+LSTM method. This section is divided into two parts. As a first step, we are interested in inspecting the behavior of the models for each individual mode. As a second step, we will evaluate the reconstructed forecast by using some of the metrics described in chapter 3 per site and per hour.

To evaluate the forecasts before reconstruction we have selected three cases shown in figures 4.4, 4.5 and 4.6. The reconstructed forecast (broken red line) and the respective observation (continuous black line) is shown in the top plot, while the lower plots show the forecast for each mode and the residue. In the mode plots

the features (obtained from a 5-day VMD) are shown with a continuous black line, and the targets (obtained from a 6-day VMD)with a broken red line. The LSTM prediction is shown with a broken red line.

The three cases were selected given they show different wind patterns. Figure 4.4 shows a day of average wind speed, figure 4.5 shows a decreasing pattern and 4.6 shows a pattern in which the wind is weak for the first 10 hours, and strong for the rest of the day.

Overall, we can see that the LSTM networks are able to provide forecasts with a frequency close to that of the nodes. The same cannot be said about the amplitude though, as in many cases the amplitude from the prediction differs significantly from the amplitude of the targets (such as u_5 in the first case and second case, and u_3 and u_4 in the third case.) For all cases, the LSTM seem to be able to continue the trend shown in the first mode in direction, although not in magnitude. The residue, which we cannot consider to be random as it seems to follow a trend, is also being approximated by the LSTM prediction.

The reconstructed forecasts shown at the top of each figure show that for cases 1 and 2 the prediction starts very close to the observations although the difference grows with time. For case 2 the prediction seems able to follow the general downward trend. On the other hand, case 3 shows a flat prediction and it does not resemble the observation.

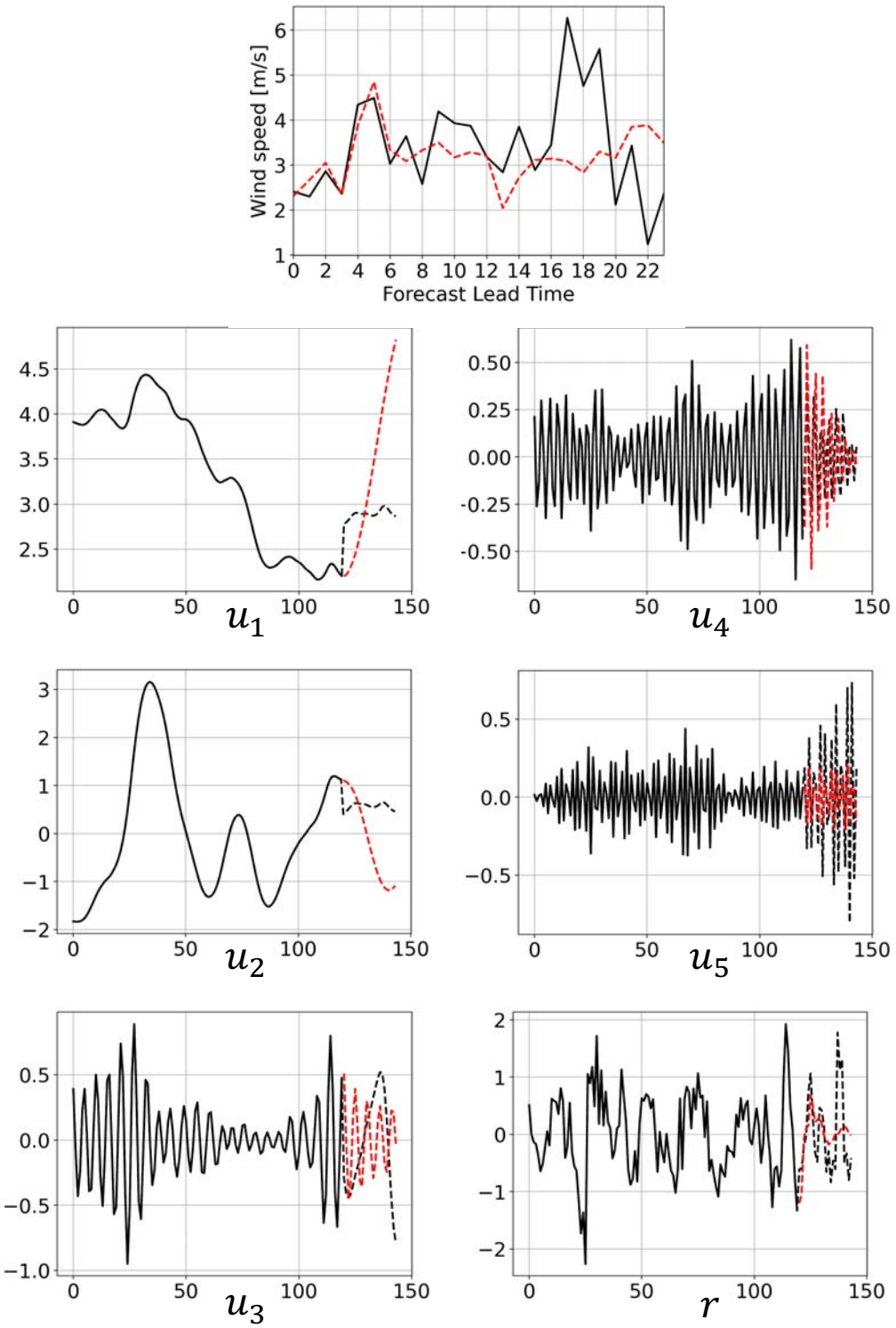

Figure 4.4: Case example 1 of 24-hour wind speed forecast using the VMD+LSTM method. The plot on the top shows the observation and the reconstructed forecast, while the lower plots show the modes that were used as inputs to each LSTM, and their corresponding predictions. Real data is shown in black, forecasts are shown in red.

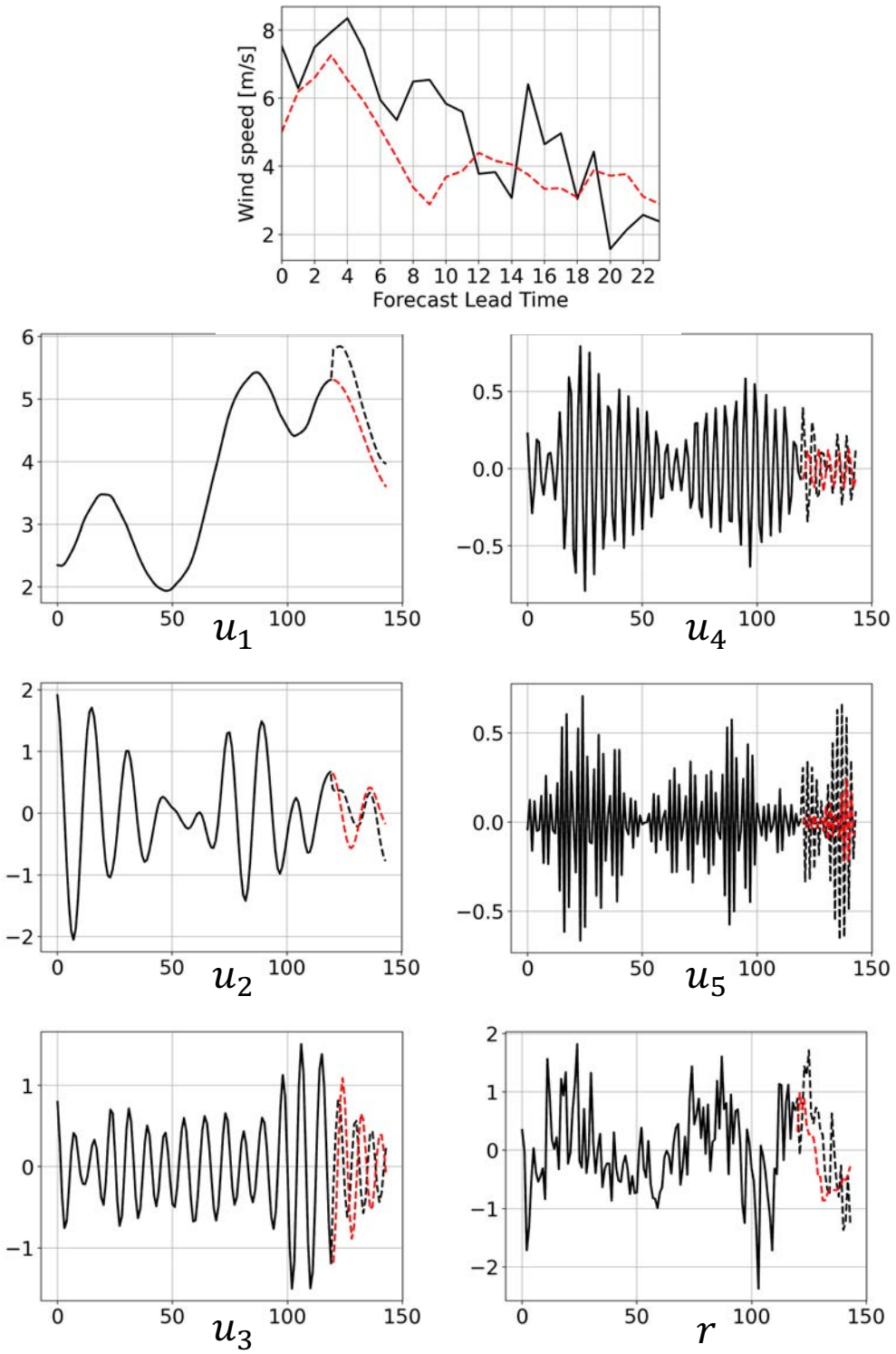

Figure 4.5: Case example 2 of 24-hour wind speed forecast using the VMD+LSTM method. The plot on the top shows the observation and the reconstructed forecast, while the lower plots show the modes that were used as inputs to each LSTM, and their corresponding predictions. Real data is shown in black, forecasts are shown in red.

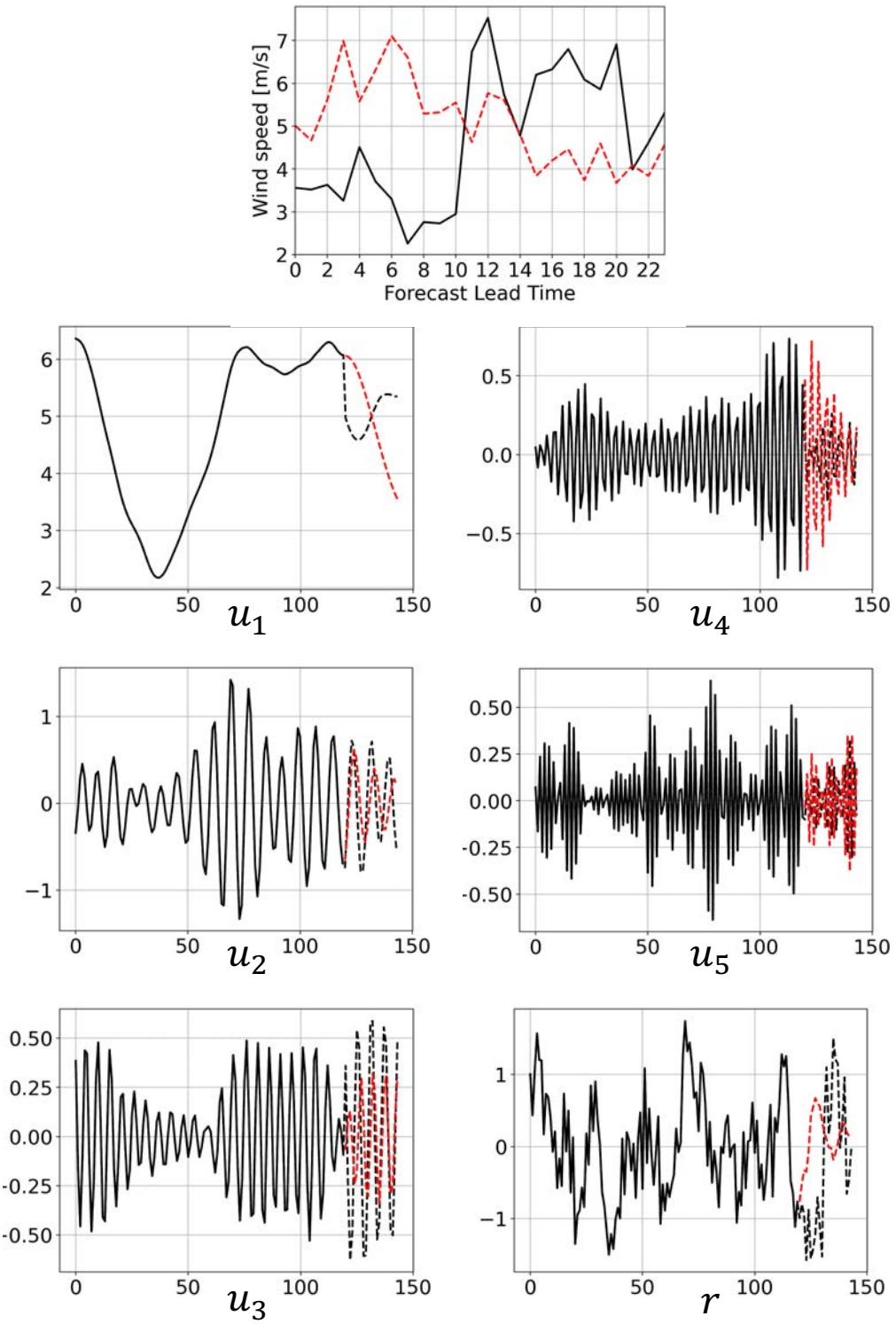

Figure 4.6: Case example 3 of 24-hour wind speed forecast using the VMD+LSTM method. The plot on the top shows the observation and the reconstructed forecast, while the lower plots show the modes that were used as inputs to each LSTM, and their corresponding predictions. Real data is shown in black, forecasts are shown in red.

We continue with the analysis of all reconstructed forecasts by using the metrics RMSE, MAE and CC. We start with results per site. Figure 4.7 shows the RMSE and MAE while 4.8 shows the CC, for all sites. The numerical data can be found in table 4.1. None of the metrics show a trend with respects of the site, as all tend to be stable. The lowest RMSE and MAE are found at site 10 with 2.339 and 1.775 m/s respectively, while the highest are found at site 8 with 2.971 and 2.301 m/s respectively. That is a difference of less than one meter per second. The CC is overall low for all sites, with the highest one at site 11 (0.29) and the lowest at sites 1 and 9 (0.238). The average metrics for all sites are 2.663 m/s for the RMSE, 2.065 m/s for the MAE and 0.261 for the CC.

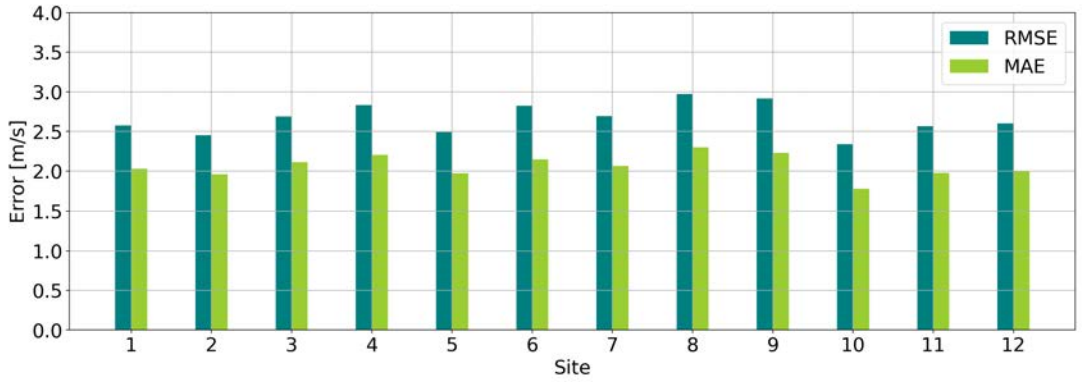

Figure 4.7: RMSE and MAE for all sites

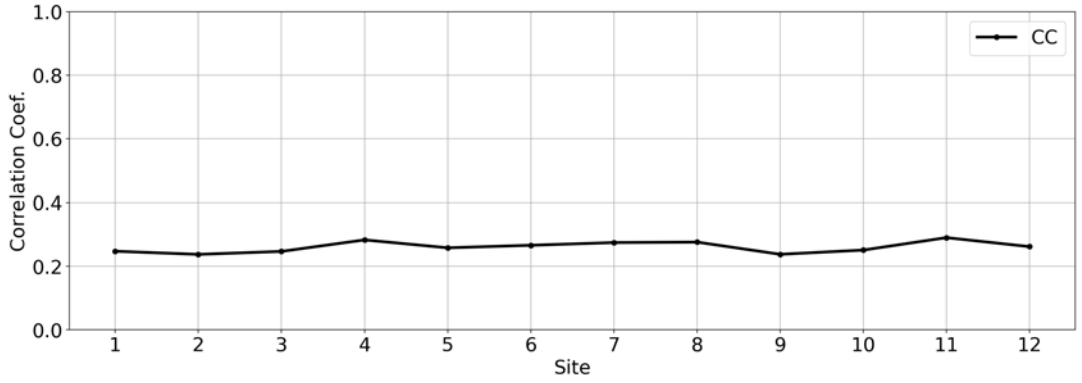

Figure 4.8: Correlation coefficient for all sites

Table 4.1: RMSE, MAE and CC results per site

Site	RMSE [m/s]	MAE [m/s]	CC
1	2.574	2.030	0.248
2	2.454	1.958	0.238
3	2.690	2.112	0.247
4	2.833	2.203	0.283
5	2.492	1.973	0.258
6	2.824	2.148	0.266
7	2.695	2.066	0.275
8	2.971	2.301	0.276
9	2.917	2.231	0.238
10	2.339	1.775	0.251
11	2.567	1.978	0.290
12	2.601	2.003	0.262

Finally, figure 4.9 and 4.10 show the same metrics but per forecast hour. The numerical data is shown in table 4.2. Here the trend is undeniable. Both RMSE and MAE start low at 1.360 and 1.025 m/s respectively. They increase rapidly and stabilize close to 3 m/s. The trend in the CC data is even more striking. The predictions show very high correlation for the first hour, but it decreases dramatically with time. From hour number 15 onwards, the correlation is practically zero.

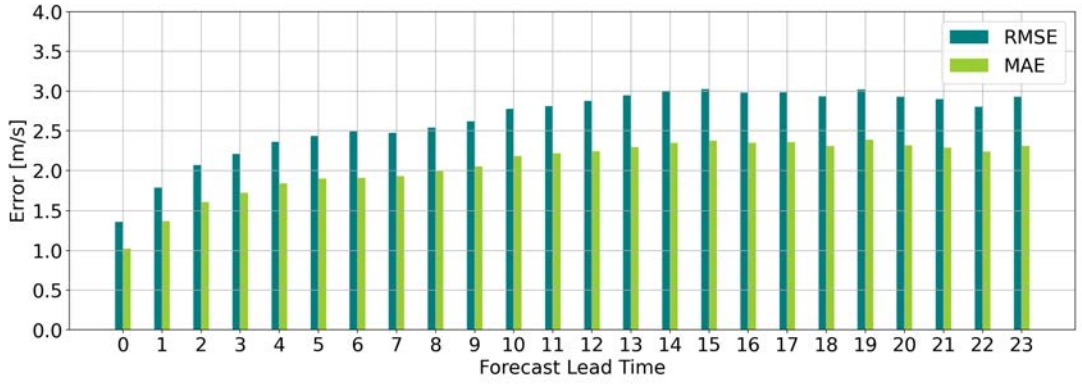

Figure 4.9: RMSE and MAE for all prediction times

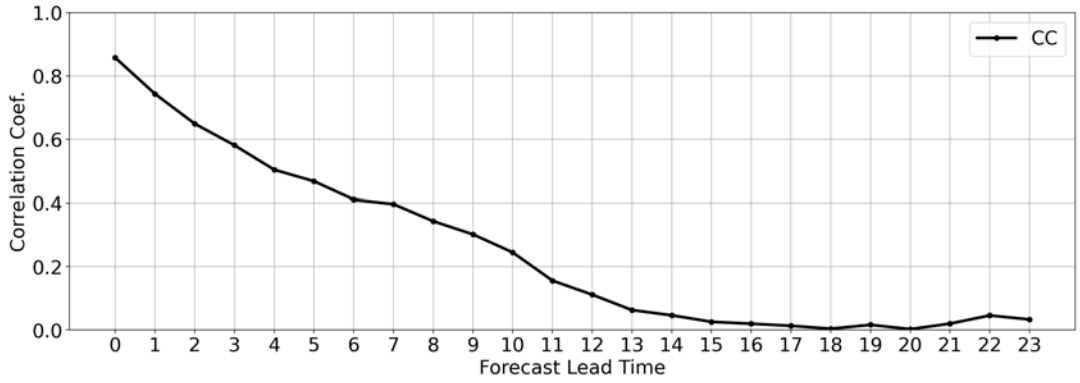

Figure 4.10: Correlation coefficient for all prediction times

Table 4.2: RMSE, MAE and CC results per hour

Forecast hour	RMSE [m/s]	MAE [m/s]	CC
0	1.360	1.025	0.858
1	1.783	1.369	0.744
2	2.067	1.610	0.649
3	2.210	1.721	0.581
4	2.362	1.839	0.504
5	2.434	1.896	0.468
6	2.492	1.908	0.411
7	2.477	1.931	0.397
8	2.541	1.991	0.343
9	2.619	2.049	0.302
10	2.775	2.181	0.245
11	2.811	2.218	0.156
12	2.876	2.242	0.112
13	2.947	2.295	0.063
14	3.003	2.347	0.046
15	3.026	2.376	0.026
16	2.980	2.349	0.020
17	2.985	2.357	0.013
18	2.934	2.309	0.004
19	3.023	2.389	0.016
20	2.931	2.317	0.003
21	2.901	2.288	0.020
22	2.802	2.239	0.046
23	2.930	2.310	0.033

4.4 Discussion

The first matter that requires discussion is the nature of the modes obtained after the VMD step, seen in figures 4.1 - 4.6. At a first glance, it can be seen that

modes u_2 through u_5 seem to show constant frequency ω_k and variable amplitude. This agrees with the expected modes described in section 4.1 were it states that the VMD will provide pure harmonics with amplitude $A_k(t)$. However, the same is not true for mode u_1. In none of the shown cases the first mode shows any zero-crossing despite having more than one local extrema. This contradicts the definition of a mode as given previously given.

Moreover, it should be mentioned that the overall decomposition looks significantly different than that shown by Dragomiretskiy and Zosso [127] and Rodrigues et al. [20]. In their work, decomposition results in modes with almost constant amplitude, or periodically variable amplitude envelope. In our case, amplitude seems to change without any discernible periodicity. While this can mean that our data is not appropriate to be treated as AM-FM signal, it can be also due to improper calibration of the VMD algorithm. Further investigation is needed to resolve the cause: either caused by the nature of our data, or by inappropriate tuning of the VMD algorithm.

When analyzing the reconstructed forecasts, it can be seen that the models show outstanding performance for the first prediction hours, but this performance decreases rapidly with time. Certainly, this is the main disadvantage derived from the present results. Similarly to statistical models, the VMD+LSTM shows good prediction skill for the first time steps, but results will *certainly* be less accurate the longer the prediction. In the case of LSTM networks, these are known to propagate the trends given as inputs, which will result in accurate short-term predictions but inaccurate long-term predictions. They initially follow the upward or downward trend of the latests time inputs, before stabilizing around the mean value of the data.

While this pure data-based approach has been shown to yield reliable results for more regular renewable sources such as solar irradiation and power (which has a daily pattern), the present results suggest that in the case of wind speed pure data-based approaches generate little-to-none value for wind speed and wind power estimation. This goes in accord with the literature presented in section 1.4. We finally argue that, as highly stochastic atmospheric process, 24-hour day-ahead wind speed prediction cannot be tackled from a pure data-based approach.

The inclusion of a dynamical model is therefore necessary. Therefore, the next chapters will explore the use of data-driven models as NWP correction methods, rather than for prediction.

4.5 Summary of this chapter

In this chapter we employed a machine learning method to obtain wind predictions from past observed data. The model in question was the long short-term memory network (LSTM) approach, coupled with a signal decomposition step provided by the variational mode decomposition (VMD) method. The intent of this model was to explore pure data-based forecasting by using machine learning methods, and establish their advantages and disadvantages for wind speed forecasting.

In this model, the wind speed time series were assumed to be a amplitude-modulated-frequency-modulated (AM-FM) signal in order to apply the VMD method. Results after decomposition suggest that this assumption might not be viable, at least for the current dataset. It might also suggest that the VMD process is highly sensitive to the decomposition parameters.

Future predictions were obtained with LSTM networks for each node, and then reconstructed to obtain a final forecast. Results showed that LSTM networks provide satisfactory results for the immediate prediction times (in this case, the first two hours), but their performance decrease radically with time. In most cases, for long prediction times the LSTM forecast will approximate the mean of the training data. We conclude that predicting wind speed based exclusively on past observations is not feasible and the inclusion of a NWP model is necessary.

Chapter 5

A deterministic multivariable neural network model

This chapter contains the proposed model for deterministic machine learning-based correction of wind speed forecasts. We start with a description of its nature and training details. Results are shown in section 5.4. A discussion about the results is offered at the end of the chapter.

5.1 Description of the model

The principal component of this first model consists mainly of a neural network. As explained in section 2.2, neural networks have a input and an output. In its most basic form, this first model is a neural network that accepts the forecast from the NWP model as its input, and produces a corrected version as an output by using the observations as our targets. Parameters must be optimized to make the network learn the mapping from NWP data to the observed data. The optimization process is described in section 2.3.3. If we denote our network as H and its parameters as $\mathbf{\Theta}$, this approach becomes

$$\hat{\mathbf{y}} = H\left(\tilde{\mathbf{y}}; \mathbf{\Theta}\right), \tag{5.1}$$

where $\tilde{\mathbf{y}} = [\tilde{y}_1, \tilde{y}_2, \cdots, \tilde{y}_{24}]$ is a 24-hour prediction from WRF model, and $\hat{\mathbf{y}}$ is the corresponding 24-hour corrected forecast. We choose to perform the correction for all 24 values at the same time, so that any temporal relations can be learned

by the network. For instance, and as with all NWP models, it is expected that the error will be greater in the latter simulation hours than in the first simulation hours. Moreover, if there are any daily periods of high, fluctuating wind where the error might be higher than the daily average, the networks would, in principle, learn to recognize it as well. Additionally, by training a different network per each site, each network should be able to learn these periods even if they occur at different times of the day in each location.

As previously mentioned in 2.2, a neural network can, theoretically, approximate any function, given that it has enough layers and nodes. However, introducing a high number of nodes is, of course, impractical due to computational and practical constraints. For this work, we choose to build our network H by taking inspiration from *denoising autoencoders*, which will be explained briefly in the next section.

5.1.1 Denoising autoencoders

Denoising autoencoders are a particular architecture of neural network which can remove noise from the input data. It consists of a neural network with an input and the output of the same size, however, with a zone of reduced nodes in the middle layers. This is called the *bottleneck* zone. The idea behind denoising autoencoders is that the network has to learn how to reconstruct the output from the input, but the bottleneck does not let all information pass through. Therefore, the network has to learn to discard unnecessary information and to let only a meaningful representation pass through. A sketch of a denoising autoencoder is shown in Figure 5.1a.

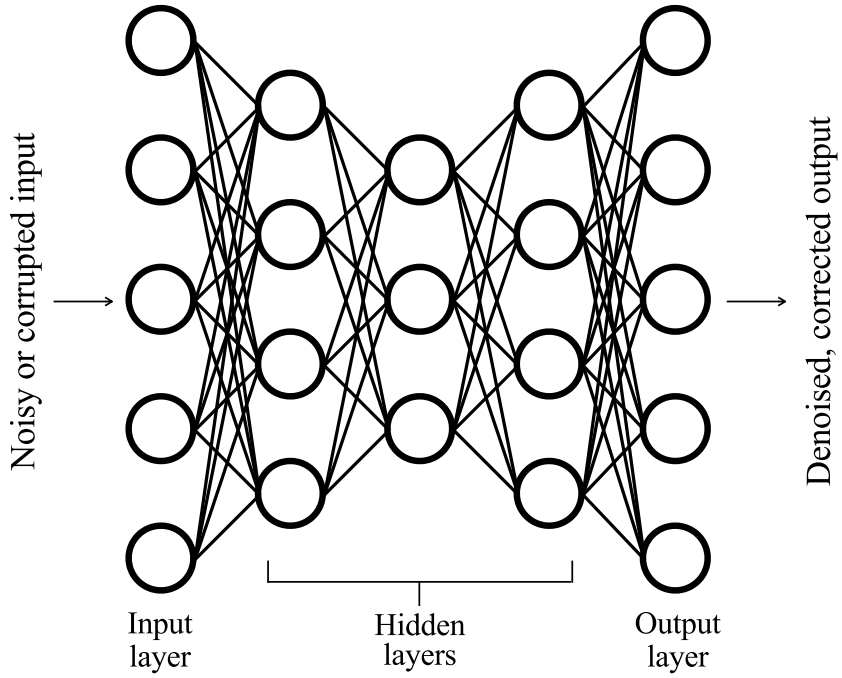

Figure 5.1: An example of a denoising autoencoder with a three-node bottleneck in the middle layer (adapted from [76]).

This mapping from corrupted to uncorrupted data can be visualized as learning a manifold [77, 131]. We assume the real information $\mathbf{x}$ lies in a abstract mathematical space, or manifold, and that the noisy information $\tilde{\mathbf{x}}$ lies outside of this manifold. Hence, the neural network learns the distribution $p(\mathbf{x} \mid \tilde{\mathbf{x}})$ and uses it to bring $\tilde{\mathbf{x}}$ to the more likely position $\hat{\mathbf{x}}$, as shown in Figure 5.2.

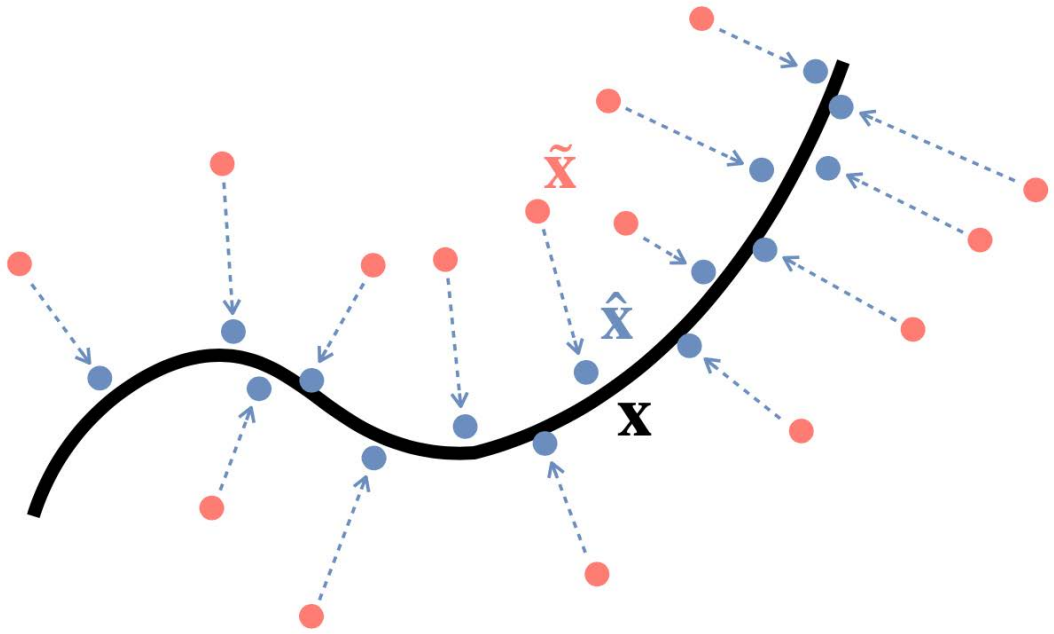

Figure 5.2: The correction process (adapted from [131]). The manifold is presented by the thick black line and the corrupted inputs $\tilde{\mathbf{x}}$ by the red dots. The autoencoder displaces each corrupted input $\tilde{\mathbf{x}}$ closer to the learned manifold. $\mathbf{x}$ and $\hat{\mathbf{x}}$ stand for the noiseless data and a reconstructed vector, respectively.

To apply the same idea to our problem, we assume that the 24-hour NWP forecast $\tilde{\mathbf{y}}$ and their corresponding observations $\mathbf{y}$ are different by a vector $\mathbf{e}$ containing all errors and biases:

$$\mathbf{y} = \tilde{\mathbf{y}} + \mathbf{e}. \tag{5.2}$$

Therefore, we consider $\tilde{\mathbf{y}}$ to be a noisy or corrupted version of $\mathbf{y}$ so that by using a network based on denoising autoencoders, the corrupted forecast can be mapped to the more accurate estimation $\hat{\mathbf{y}}$.

Although the idea of denoising data with neural networks was originally introduced by Le Cun and Fogelman-Soulié [132] and Gallinari et al. [133], modern denoising autoencoders (as presented by Vincent et al. [131] and described in this section) were developed with the distinct intent of finding and extracting meaningful features from data. Naturally, modern denoising autoencoders have been used on actual image and audio denoising tasks [134, 135].

5.1.2 Architecture

For this first deterministic approach we will present two networks. First, we introduce the network shown in Fig. 5.3 with an input and output size of layers of 24 neurons. The only hidden layer serves as the bottleneck with 23 nodes. Before reaching this final design, other prototypes were tested with different numbers of layers and nodes. These are described in section 5.6.

The input for this network is the wind speed $\tilde{s}$ from the NWP model, and we refer to the input of the network as a vector $\tilde{\mathbf{s}}$ of size 24. Since the input consists of a single variable, we call this network *single variable neural network NN-S* where 's' stands for 'speed'. All layers are fully connected, thus, the output z_n from the nth node in the hidden layer is

$$z_n = \phi\left(\tilde{\mathbf{s}}^T \mathbf{w}_n + b\right), \tag{5.3}$$

where ϕ is an activation function, θ_n are the weights connecting the input nodes and node n, and b is a bias term. The chosen activation functions for the hidden layer and the output layer are ReLU and sigmoid respectively,

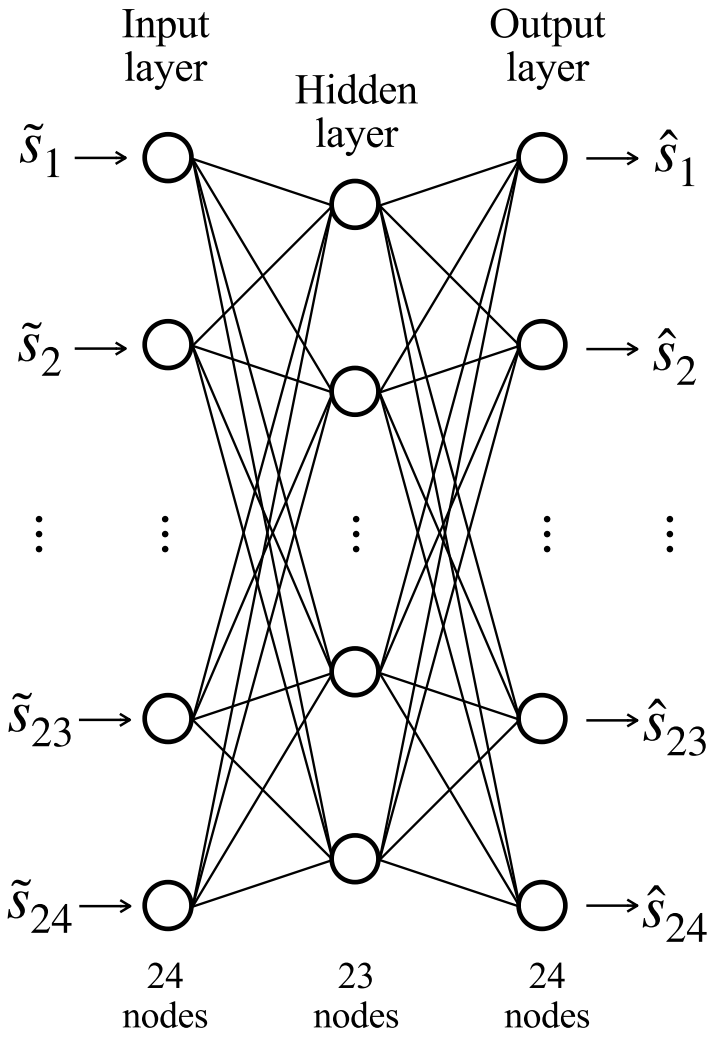

Figure 5.3: NN-S: A wind speed correction neural network in which the input is the NWP wind speed forecast

We hypothesize that feeding additional information about the state of the atmosphere into the network will greatly enhance its correction skill. With this idea in mind, we present a second network where the input consists of the 24-hour wind speed forecast $\tilde{\mathbf{s}}$, and also the forecasts for other atmospheric variables that are correlated to wind speed, namely wind direction $\tilde{\mathbf{d}}$, pressure $\tilde{\mathbf{p}}$ and temperature $\tilde{\mathbf{t}}$. These extra input variables were chosen on the assumption that they are correlated to the variable of interest [49]. Moreover, this same set of variables is used in other prediction/correction methods where a more comprehensive state of the atmosphere is needed as input [49, 50] The predictions of these variables are

extracted from the same WRF model run as the wind speed prediction, for the same time period.

The architecture of this network is shown in Figure 5.4. It has an input layer with 96 nodes for the 24-h forecast of the four variables, two hidden layers with 24 and 23 nodes respectively, and an output layer with 24 nodes. The most important feature in this network is that the first hidden layer is not fully connected to the input layer, but rather locally connected. Each node in the first hidden layer is connected only to the input nodes that receive the values for a particular hour. In other words, the nth node in the hidden layer is connected only to the input nodes for the predicted wind speed $\tilde{s}$, wind direction $\tilde{d}$, pressure $\tilde{p}$ and temperature $\tilde{t}$ at the nth hour. Thus, the output z_n from the nth node in the hidden layer becomes

$$z_n = \phi\left(\theta_{s,n}\tilde{s}_n + \theta_{d,n}\tilde{d}_n + \theta_{p,n}\tilde{p}_n + \theta_{t,n}\tilde{t}_n + b\right). \tag{5.4}$$

During training the connecting weights will be updated according on the contribution of each input variable to the correction process. With this architecture we expect the first layer to focus exclusively on the interaction between the input atmospheric variables that might contribute to the correction. Subsequent layers can then focus on the temporal relations between the inputs. As with the previous network, we show the rationale of choosing this architecture in section 5.6, were we compare with other layouts. We denominate this network *multi variable neural network NN-SDPT*, where "SDPT" stands for speed, direction, pressure and temperature.

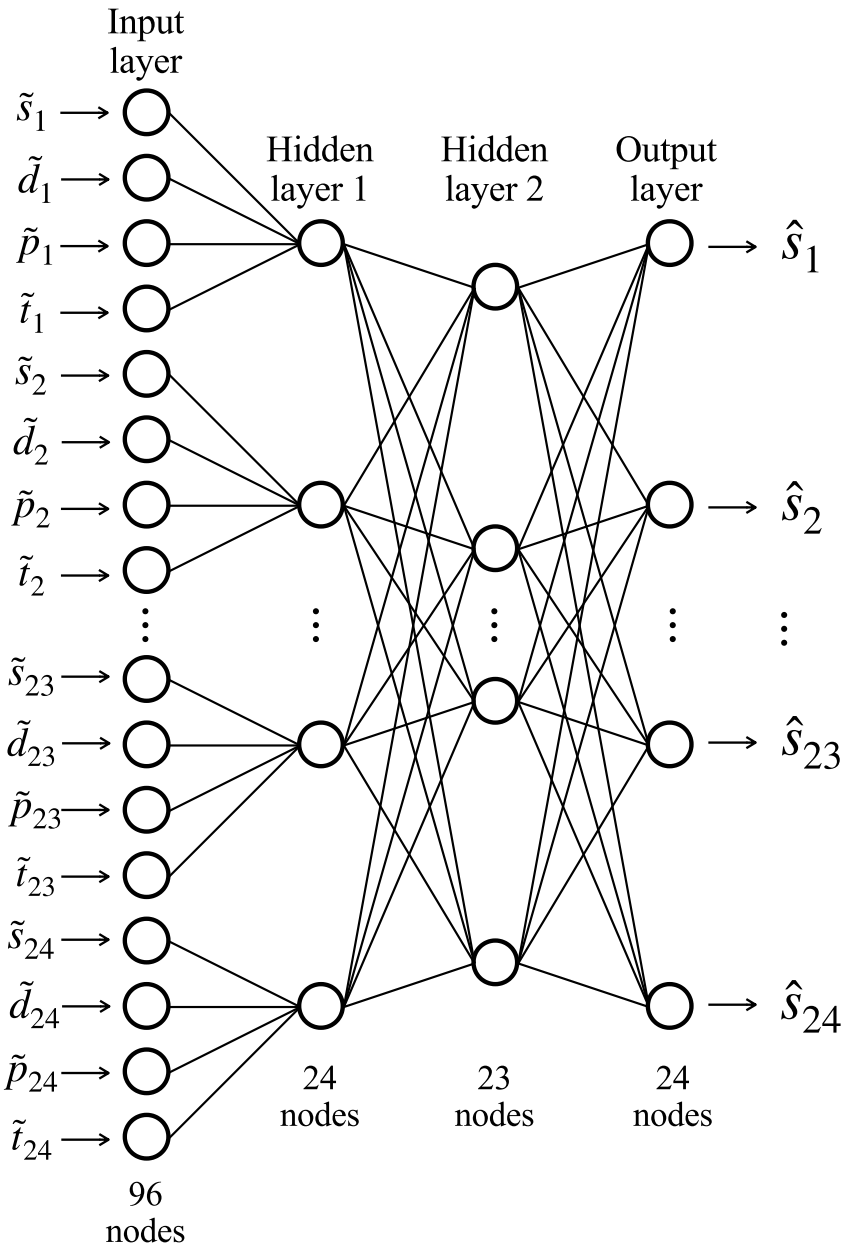

Figure 5.4: NN-SDPT: A wind speed correction neural network in which the inputs are the WRF-predicted wind speed, wind direction, pressure and temperature.

Finally, table 5.1 shows the details for both architectures, including their input output layers and activation functions.

Table 5.1: Description of the neural networks presented in this study.

Network	Input (from NWP)	Output	Architecture	Activation Functions
NN-S	24-h wind speed forecast	Corrected wind speed forecast	1. Input layer (24 nodes) 2. Hidden layer (23 nodes, fully-connected with previous layer) 3. Output layer (24 nodes, fully-connected with previous layer)	ReLU in all hidden nodes Sigmoid in all output nodes
NN-SDPT	24-h wind speed, direction, pressure, and temperature forecasts	Corrected wind speed forecast	1. Input layer (96 nodes) 2. Hidden layer 1 (24 nodes, locally-connected with previous layer) 3. Hidden layer 2 (23 nodes, fully-connected with previous layer) 4. Output layer (24 nodes, fully-connected with previous layer)	ReLU in all hidden nodes Sigmoid in all output nodes

5.2 Training details

For this model we employ the dataset from the Awaji Island wind farm, as described in chapter 3. Following the practices described in section 2.3, the dataset was partitioned into two: a training and validation set, and a test. Each set contained data between the following time periods:

1. Training and validation: 2016-01-03 to 2019-12-24 (90% of data)
2. Testing: 2019-12-25 to 2020-06-01 (10% of data)

Training and validation were carried out in a 10-fold cross-validation fashion and the best fold was taken for each site. A different network is trained for each site so that each network can learn local patterns or intricacies in the wind behavior that are important for the correction process. A total of 30 (15 NN-S and 15 NN-SDPT) networks are trained, all of them with ADAM as optimization (section 2.3.3) and mean square error as the loss function. All inputs are scaled by using normalization (section 2.3.2 before feeding them to the networks.

5.3 Baseline method

The baseline method for this chapter is the Model Output Statistics (MOS) method, introduced by Glahn and Lowry [43]. MOS is a family of techniques

that incorporate dynamical forecast information into statistical weather forecasts. This is done by finding statistical relationships between the NWP's output and observed and/or additional model data.

For the MOS implementation in this work, a multi-variable linear regression model is employed. Given a deterministic NWP prediction at present time ($t = 0$), past predictions and observations for the same location and lead time are recalled from a history of past predictions observations. The regression model is built by the recalled forecasts as predictors and their corresponding observations as predictands. The corrected value (MOS prediction) is obtained by using the regression model with the NWP forecast at $t = 0$ as predictors. Figure 5.5 shows a schematic of the process.

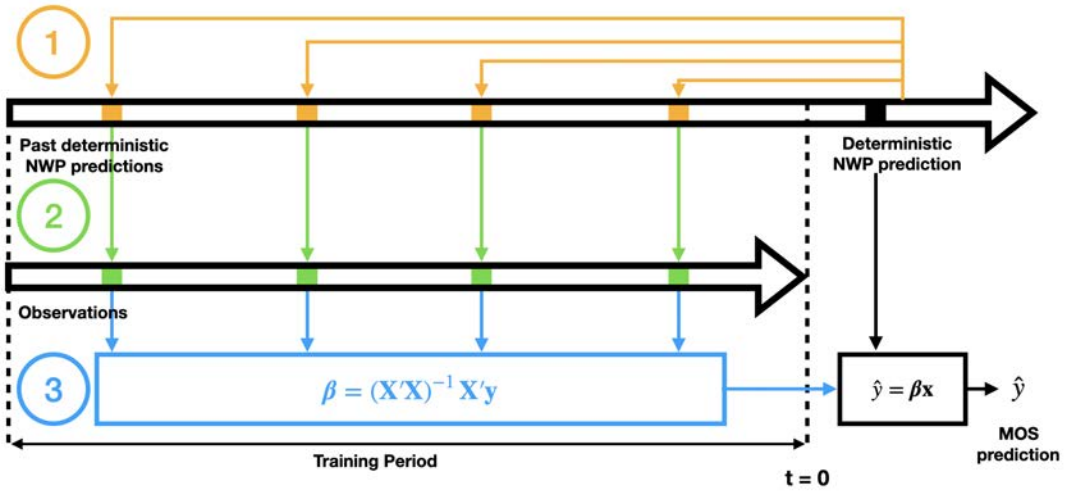

Figure 5.5: Schematic of the MOS method (adapted from [50]).

For our MOS implementation, the predictors are the same four weather variables as in NN-SDPT. The MOS model was trained by using 1 year of past data.

5.4 Results

In this section we present results for all trained networks. We employ the metrics and tools described in chapter 3 for deterministic models, these are the root mean square error (RMSE), mean absolute error (MAE), correlation coefficient (CC) and Taylor diagrams. We also show portions of the time series of two of the sites to see the actual improvement of the models over the NWP data and the baseline

method. All the results shown in this work correspond to the testing period from 2019-12-25 to 2020-06-01.

We start with the RMSE, MAE and CC metrics, shown in Tables 5.2, 5.3 and 5.4respectively. The RMSE is also shown graphically in Figure 5.6.

The "Raw" label corresponds to the parameter calculation between the observations versus the uncorrected WRF wind speed forecasts, while "NN-S" and "NN-SDPT" correspond to the observations versus the forecasts corrected with each method respectively. The percentage of improvement relative to the raw forecast is also shown for each parameter.

Figure 5.6 shows the RMSE values before and after correction, as well as the percentage of improvement. For all 15 sites, both networks are able to reduce the RMSE in the testing period.

The parameters MAE and CC are also improved in all 15 sites. Furthermore, all metrics regarding the correction with the multi-variable network, i.e., NN-SDPT, are superior to those of the single variable NN-S.

For NN-S specifically, the percentage of improvement of RMSE ranges from 6.86% in site 5 to 17.63% in site 1, with an average of 11.55%. For NN-SDPT, the relative improvement increases with values ranging from 8.76% in site 4 to 26.55% in site 15, with an average of 16.52%.

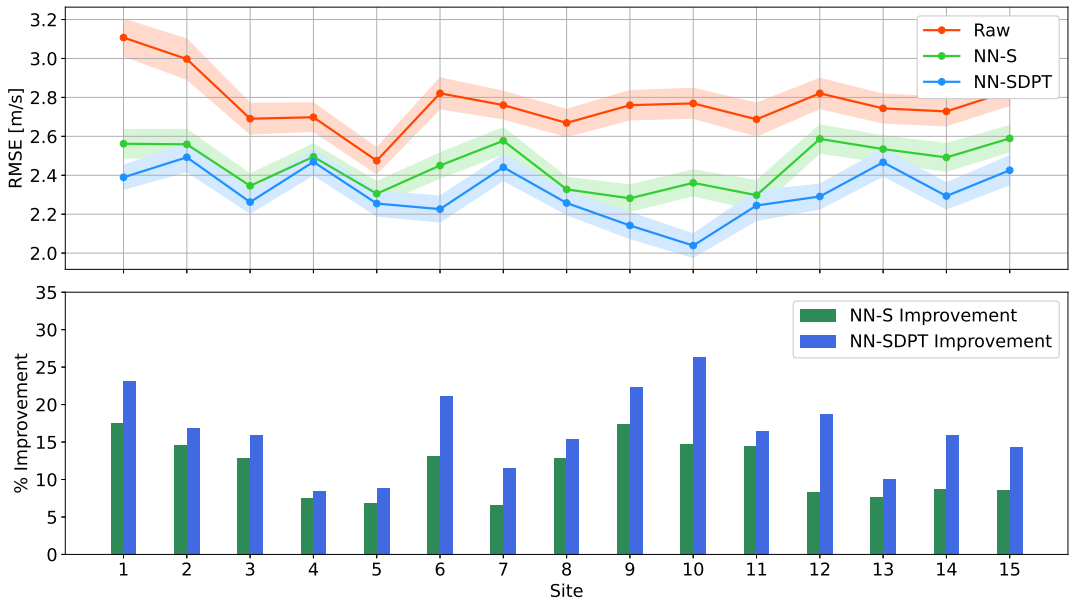

Figure 5.6: RMSE values for the WRF output (Raw) and after the correction with networks NN-S and NN-SDPT for each site. The shaded areas represent their respective 95% confidence intervals.

Table 5.2: The RMSE values [m/s] for all 15 sites, before ("Raw") and after (i.e., "NN-S" and "NN-SDPT") correction.

Site	Raw	NN-S	% Improv. NN-S	NN-SDPT	% Improv. NN-SDPT
1	3.11	2.56	17.63%	2.38	23.48%
2	3.00	2.56	14.61%	2.49	16.83%
3	2.68	2.34	12.78%	2.26	15.85%
4	2.70	2.50	7.38%	2.47	8.76%
5	2.47	2.30	6.86%	2.25	8.93%
6	2.86	2.45	14.11%	2.23	22.06%
7	2.78	2.59	6.88%	2.45	11.82%
8	2.67	2.33	12.80%	2.26	15.41%
9	2.75	2.27	17.39%	2.14	22.38%
10	2.77	2.35	14.94%	2.03	26.55%
11	2.69	2.30	14.64%	2.24	16.63%
12	2.81	2.58	8.29%	2.28	18.77%
13	2.74	2.53	7.68%	2.46	10.16%
14	2.72	2.48	8.65%	2.29	15.91%
15	2.85	2.61	8.62%	2.45	14.26%
Average			11.55%		16.52%

Table 5.3: The MAE values [m/s] for all 15 sites, before ("Raw") and after (i.e., "NN-S" and "NN-SDPT") correction.

Site	Raw	NN-S	% Improv. NN-S	NN-SDPT	% Improv. NN-SDPT
No.1	2.40	2.00	16.47%	1.87	21.85%
No.2	2.27	2.00	11.92%	1.94	14.48%
No.3	2.04	1.86	8.64%	1.79	11.94%
No.4	2.10	1.97	6.14%	1.94	7.56%
No.5	1.92	1.81	5.55%	1.75	8.77%
No.6	2.22	1.94	12.56%	1.73	22.09%
No.7	2.18	2.06	5.52%	1.90	12.89%
No.8	2.09	1.83	12.50%	1.76	15.78%
No.9	2.15	1.74	19.34%	1.65	23.33%
No.10	2.15	1.80	16.24%	1.56	27.72%
No.11	2.06	1.73	16.22%	1.68	18.38%
No.12	2.17	1.99	8.09%	1.76	18.56%
No.13	2.10	1.96	6.70%	1.89	9.76%
No.14	2.10	1.92	8.64%	1.76	16.27%
No.15	2.24	2.09	6.64%	1.89	15.37%
Average			10.74%		16.32%

Table 5.4: The CC values [m/s] for all 15 sites, before ("Raw") and after (i.e., "NN-S" and "NN-SDPT") correction.

Site	Raw	NN-S	% Improv. NN-S	NN-SDPT	% Improv. NN-SDPT
No.1	0.62	0.74	18.88%	0.78	25.31%
No.2	0.60	0.70	15.73%	0.72	19.79%
No.3	0.70	0.73	5.29%	0.75	8.19%
No.4	0.71	0.75	5.82%	0.75	6.92%
No.5	0.74	0.78	5.22%	0.79	6.50%
No.6	0.68	0.74	8.58%	0.79	16.03%
No.7	0.70	0.75	6.94%	0.77	9.81%
No.8	0.69	0.75	7.59%	0.77	10.48%
No.9	0.68	0.74	8.93%	0.78	14.66%
No.10	0.67	0.72	7.40%	0.79	18.75%
No.11	0.69	0.75	7.74%	0.77	10.72%
No.12	0.68	0.72	6.83%	0.78	15.70%
No.13	0.70	0.75	7.04%	0.77	8.88%
No.14	0.66	0.68	3.50%	0.75	13.16%
No.15	0.68	0.73	7.73%	0.77	13.15%
Average			8.21%		13.20%

Figure 5.7 shows the RMSE values calculated for each of the 24 simulated hours accumulated for all sites. Here the baseline method MOS is included as reference (orange dashed line). The RMSE for the raw predictions (solid red line) is at its minimum during the first hours of the simulation, and it grows steadily until its peak at the 12th simulation hour (20:00 JST). After this period, the error decreases slightly before increasing again during the last hours of the simulation. The RMSE for NN-S and NN-SDPT is shown with the green and blue lines, respectively. The error is largely decreased for all hours and roughly in similar proportions. The same increase-decrease-increase pattern is present. Moreover, the multivariable NN-SDPT again outperforms NN-S at all times. The difference in performance is more evident during the last hours of the simulation.

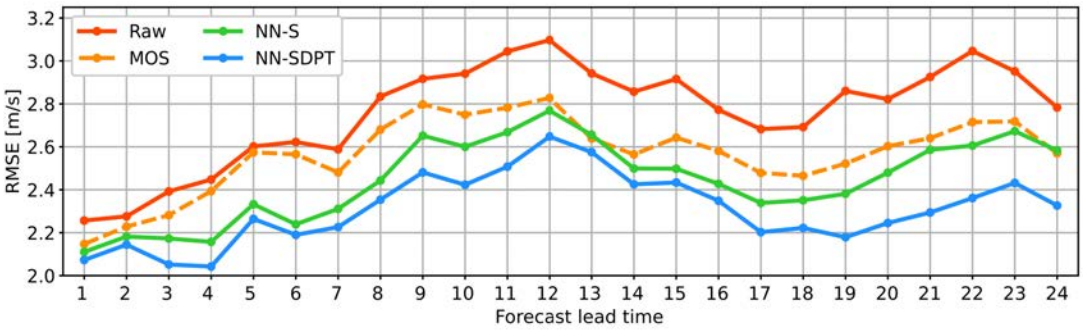

Figure 5.7: RMSE values for the WRF output (Raw) and after the correction with MOS, NN-S and NN-SDPT, per hour

We use the Taylor diagram to obtain a graphical interpretation of the results. Figure 5.8 shows the Taylor diagram for all four datasets, accumulated for all sites and all hours. All three correction models result in a favorable increase in correlation and decrease in CRMSE, however at the same time all models show an undesirable decrease in standard deviation. The decrease in standard deviation is most significant for MOS, followed by the single variables NN-S and then the multi variable NN-SDPT.

Figure 5.9 displays the individual positions for all 15 sites, for the raw forecast and the neural network models. As already seen in the overall Taylor diagram (Fig. 5.7), after correction with NN-S the markers tend to move towards zero in the standard deviation axis. This means that a great amount of variability in the data is lost during the correction. However, this displacement is reduced when

correcting with NN-SDPT, as the markers seem to move in opposite direction. This results in an even higher reduction in CRMSE and a higher increase in the correlation coefficient.

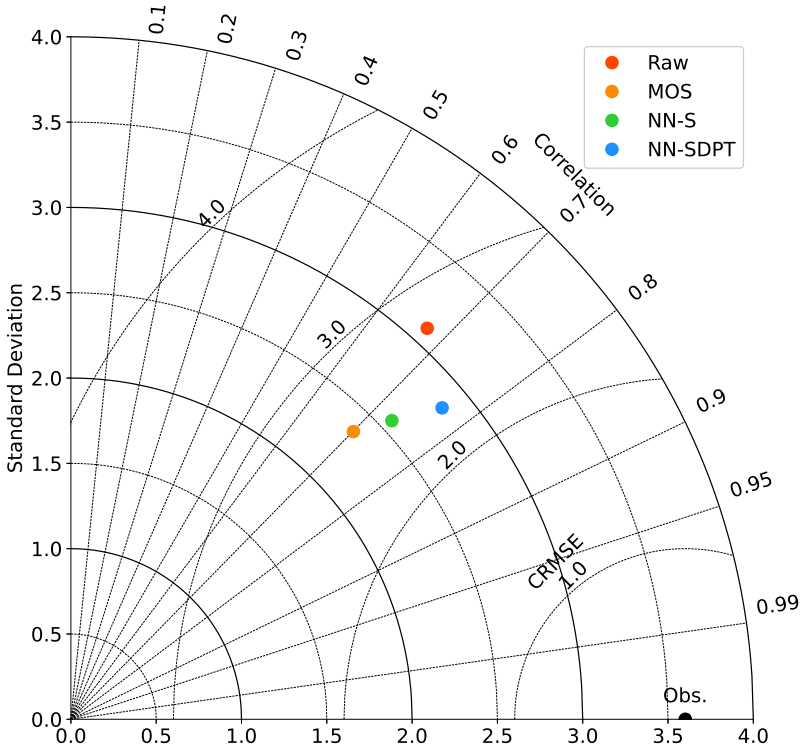

Figure 5.8: Taylor diagram for the raw forecast, MOS correction, and the NN-S and NN-SDPT correction methods, for all 15 sites.

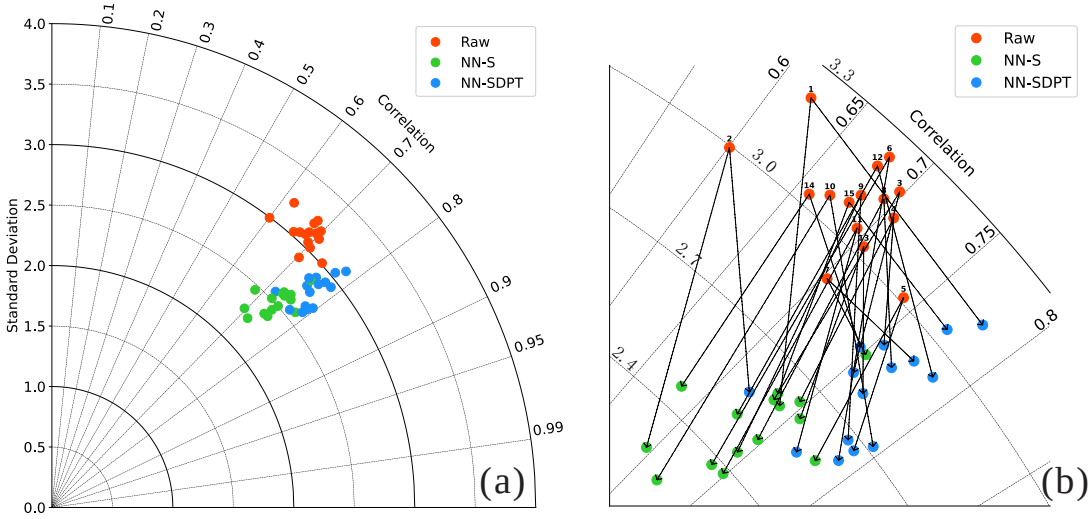

Figure 5.9: (a) Taylor diagram for all 15 sites, showing markers for before and after the correction with NN-S and NN-SDPT. (b) An enlarged portion of (a) showing that NN-S tends to displace the marks towards the right, while NN-SDPT tends to displace the mark towards the left.

Finally, in Figure 5.10 we present five examples of the corrected and uncorrected wind speed time series, together with the observed values, for a period of seven days. The metrics for these examples are shown in Table 5.5. These cases were selected since they show instances of the WRF model overestimating and/or underestimating the wind speed that the neural networks were able to correct up to some point.

In cases A and D, the raw forecast incorrectly predicts brief periods (shorter than a day) of high-speed wind. The NN-SDPT correction method successfully corrects this overestimation in case A and is able to bring the values closer to the observations in case D.

These cases highlight as well the superior performance of network NN-SDPT. In the latter hours of case A, the NN-SDPT correction is clearly superior to the NN-S correction. In case B, another high-speed period lasting approximately three days can be partially corrected only by NN-SDPT. In cases C and E, the raw, uncorrected values are already very close to the observations, however, NN-S actually degrades the forecast, especially during the high wind speed periods. In case C, the NN-S correction is actually worse than the raw prediction, as it can be seen in the RMSE values shown in Table 5.5.

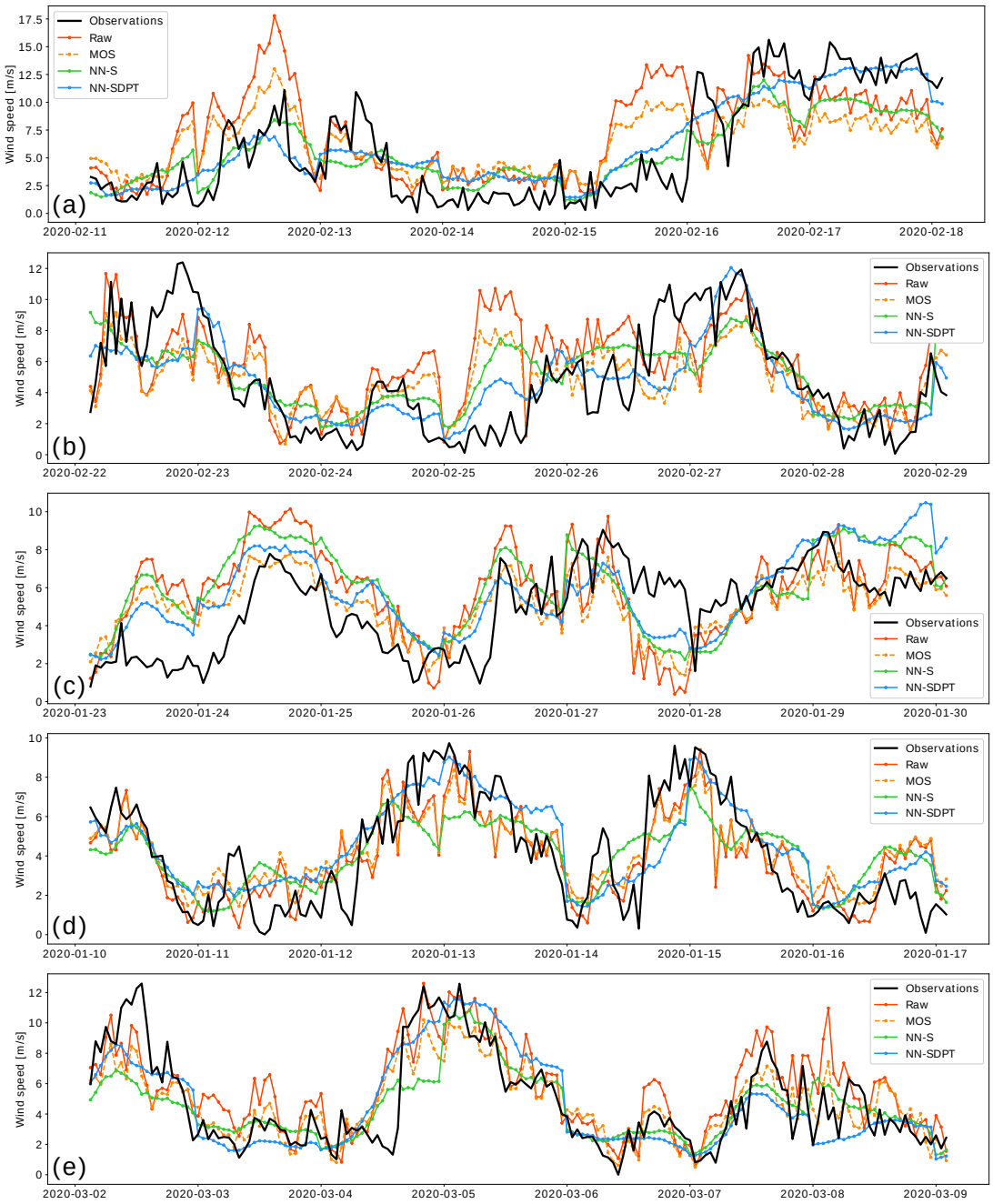

Figure 5.10: Wind speed observations and forecasts (corrected and uncorrected) for a period of seven days, for five selected cases. (a)–(e) represent the Case A–E displayed in Table 5.5.

Table 5.5: Case description and the RMSE values [m/s] for those cases before and after correction.

Case	Site	Time period	Raw	MOS	Improv. MOS	NN-S	% Improv. NN-S	NN-SDPT	% Improv. NN-SDPT
A	No.1	2020-02-11 – 2020-02-18	4.36	3.92	10.09%	2.72	37.65%	2.22	49.19%
B	No.9	2020-02-22 – 2020-02-29	3.24	2.86	11.73%	2.58	20.28%	2.13	34.29%
C	No.3	2020-01-23 – 2020-01-30	2.65	2.02	23.77%	2.47	6.86%	2.03	23.89%
D	No.5	2020-01-10 – 2020-01-17	1.78	1.84	-3.37%	1.98	-11.40%	1.65	6.94%
E	No.10	2020-03-02 – 2020-03-09	2.18	1.87	14.22%	2.02	7.68%	1.87	14.22%

5.5 Discussion

Generally speaking, the output from NN-S and NN-SDPT shows better evaluation metrics when compared to the raw forecast and the baseline method. Regarding the RMSE reduction, this is to be expected since the loss function used to train the network was the mean square error.

Arguably the most important point that can be made is that NN-SDPT is better than NN-S in most if not all metrics, meaning that the architecture designed to extract and assimilate additional information from wind direction, pressure and temperature forecasts is working as intended, and it is indeed contributing in correcting the wind speed variable.

It stands out that, although the models can produce corrected forecasts with reduced error, the percentage of improvement varies considerably between sites. The lowest improvement percentage for both networks occurs at sites 4, 5, and 13. A closer look at Figure 5.6 and Tables 5.2, 5.3 and 5.4 reveals that the raw predictions for these sites are much better than other sites (e.g., site 5 has the lowest raw RMSE and MAE, and the highest CC). Therefore, it can be implied that in these sites there is little room for correction when compared to the other sites.

Regarding the evolution of the error with respect to simulation time (Fig. 5.7), an increase-decrease-increase pattern is present in the raw error, the MOS error and both neural network models. As mentioned previously in section 3.1 and shown in figures 3.5 and 3.9 there is a period of strong winds that extends from the 9th hour (17:00 JST) to the 12th (20:00 JST). A per-site analysis revealed that this strong wind period exists in all sites, although it is more marked in some sites than in others. We argue that this is the cause of the pattern in Figure (Fig. 5.7) since naturally stronger wind will lead to a more pronounced error. The final increase in error during the last hours of the simulation could be attributed to the normal decrease in performance of the NWP model after long simulation times.

The main takeaway from the Taylor diagrams is that correction results in a decrease in the standard deviation of the results, meaning that the corrected data shows less variability than the observations, and even the NWP results. This can also be seen in the time series plots, as the results for the neural network models are flatter. Fluctuations in wind speed that might be correctly predicted by the NWP model, might be flattened out by the correction process. Once again, this can be attributed to the choice of RMSE as the loss function. As it will be shown in section 6.1, the RMSE is a composite metric containing both bias and variance, and minimizing one will inevitable increase the other. Nevertheless, conserving information about the variance of a stochastic phenomena such as wind is critical for its understanding. The loss of variability in this model is one of the main differences with the probabilistic model that will be introduced in the next chapter.

As a final comment we should remark that the neural networks as such do not show any decrease in correction performance with respect of simulation time. The difference in correction skill per hour shown in figure 5.7 is attributed to the strong winds and the intrinsic NWP error increase. This is in stark contrast with the pure data-based VMD+LSTM model introduced in chapter 4, where the prediction performance decreases sharply with time. Results so far suggest that data-driven methods in conjunction with NWP models are a much more viable approach for wind speed forecasting.

5.5.1 Analysis of NN-SDPT

Neural networks are usually referred as black box models, meaning that the internal working of the model is assumed to be "inaccessible" and the only concern are the inputs and the accuracy of its output. This is mostly due to the difficulty of interpreting the final weights of a network after training. Nevertheless, having a look at the final weights can help determine which features contribute more to the network's output. This kind of analysis can also provide new insights into complex, non-linear physical systems [136]. In this section, we attempt to study the internal state of the network NN-SDPT, particularly the connections between the input and the first hidden layer. As shown in Eq. 5.4, each node in the first hidden layer contains a weight for each atmospheric variable. The higher the magnitude of the weight, the higher the respective variable contributes to the correction process.

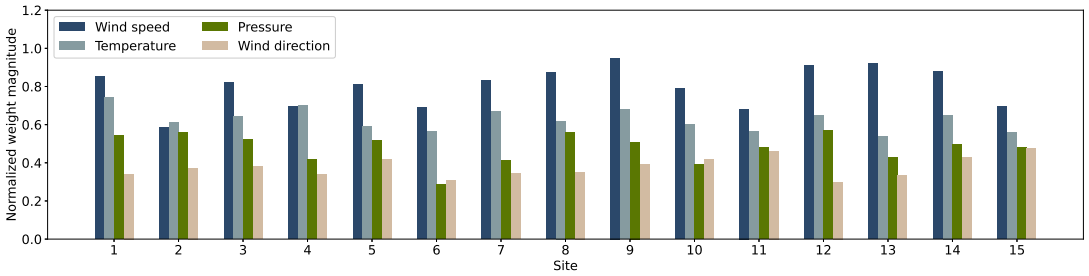

Figure 5.11: Normalized weights from the first hidden layer of the NN-SDPT network, for all sites. The values shown correspond to the average of the 24 weights for each variable in the hidden layer.

In order to study the contribution of each variable, we have extracted the final weights from the links between the input and first hidden layer. These are shown in Figure 5.11 after normalization and averaging for each site. As expected, wind speed has the highest weight in the correction process for all sites except for sites 2 and 4. This means that wind speed is the variable that the network finds the most useful for the correction of wind speed. Although this might seem trivial and obvious from our perspective, it should be mentioned that the network has no previous knowledge of which variable is being input in each node, and it knows no physical information about the atmospheric processes. The fact that

the network precisely figured out the variable that was being corrected suggests that the correction is not happening by pure chance.

In almost all sites, wind speed is followed by temperature, pressure, and finally, wind direction. It should be noted that this does not necessarily mean that there is a higher correlation between wind speed and temperature than with the other variables. It simply means that the model has found useful information in the temperature time series that improves the correction of the wind speed time series. Although there might be deeper connections and relations in the subsequent layers that we are not taking into account, this brief analysis shows that the correction in each site depends on each variable with different magnitudes, suggesting that each site is certainly subject to very different wind patterns. However, it should be mentioned that any attempt at understanding the weights of a neural network can only be done speculatively, as there can be different interpretations of their meanings [136].

5.6 Tests during architecture design

Before obtaining the final architecture for the multivariate network NN-SDPT, several network architectures were evaluated. Major variables in the design include number of layers, number of nodes (particularly in the bottleneck layer), addition of regularization methods and, most importantly, the effect of a partially connected layer versus a fully connected layer. Once the final multivariate network NN-SDPT was obtained, the univariate network NN-S was straightforwardly created as a univariate version of NN-SDPT. This section summarizes such tests.

All candidate architectures were trained for all 15 sites using 10-fold cross validation. Early-stopping was applied with a stopping criteria of minimum validation loss and a patience hyperparameter of 200 epochs 2.3.1.

The main criteria to evaluate an architecture in this section is the average RMSE across all 10 validation folds, and the corresponding improvement (reduction) in RMSE when compared with the raw values.

Locally-connected layers vs fully-connected layers

In this subsection we justify our rationale of using a locally-connected layer for the multivariable network NN-SDPT. Table 5.6 shows the average improvement in RMSE for each site for two networks: the original NN-SDPT and a modified version with a fully-connected layer instead. On the other hand, table 5.7 shows the results for locally and fully connected networks but with 24 nodes in the bottleneck layer (no reduction). It the locally-connected version performs better in 12 out of 15 sites for the 23 node case and 9 out of 15 sites for the 24 node case. The overall average improvement is greater as well in both tables.

In the case of the networks in table 5.6, a Wilcoxon signed-rank significance test was carried out. Using all available pairs (15 sites and 10 folds, n=150). The test concluded in that there is a significant difference between the RMSE of both networks ($z=-2.60$, $p<.01$). The same test was also carries out for CC and MAE, both showing significant differences as well ($z=2.04$, $p<.05$ and $z=-2.18$, $p<.05$ respectively).

Effect of the number of nodes in the middle layer

Here we present the results for networks with different node numbers in the middle bottleneck layer. For the following tables we have omitted the average RMSE and we only show the improvement percentage.

Table 5.8 shows versions of NN-SDPT where the middle layer was changed to have from 22 to 25 nodes. In this case, the 22 node version shows a greater overall average improvement than the 23node version. However, the 23 node version shows superior improvement on 7 out of 15 sites, and it was chosen on that basis.

Effect of an extra layer

Table 5.9 shows results comparing NN-SDPT with a modified version in which an extra 24-node layer was included. The results clearly show that the addition of this layer decreases the correction skill significantly.

Effect of dropout regularization

Table 5.10 shows results comparing NN-SDPT with a modified version where a dropout layer was included before the las layer. Two dropout rates were tested, 0.1 and 0.2. The results show that using a rate of 0.1 might help in some sites, but still no dropout is better for 9 out of the 15 sites, as well as for the overall average. A rate of 0.2 offers no improvement at all.

Another Wilcoxon signed-rank significance test was conducted between the network without dropout and the one with rate = 0.1. The test concluded that there is a significant difference between their RMSE values (z=-1.99, p<.05), CC values (z=2.10, p<.05) and MAE values (z=-4.44, p<.01).

Table 5.6: Average RMSE and improvement for fully and locally-connected networks with 23 nodes in the middle layer

Site	Raw RMSE	Locally-connected 23 nodes in the middle layer (Original NN-SDPT)		Fully-connected 23 nodes in the middle layer (Modified NN-SDPT)	
		Average RMSE	Average Improvement	Average RMSE	Average Improvement
1	3.049	2.266	25.687%	2.255	**26.002%**
2	2.858	2.244	**21.481%**	2.254	21.079%
3	2.651	2.168	**18.233%**	2.182	17.673%
4	2.627	2.248	14.445%	2.244	**14.612%**
5	2.564	2.243	12.516%	2.241	**12.595%**
6	2.728	2.020	**25.931%**	2.023	25.839%
7	2.722	2.341	**14.024%**	2.361	13.297%
8	2.594	2.056	**20.731%**	2.079	19.842%
9	2.703	2.050	**24.137%**	2.072	23.354%
10	2.692	1.921	**28.630%**	1.934	28.146%
11	2.645	2.203	**16.682%**	2.228	15.734%
12	2.623	2.088	**20.394%**	2.105	19.773%
13	2.633	2.267	**13.890%**	2.297	12.731%
14	2.652	2.100	**20.806%**	2.114	20.286%
15	2.722	2.307	**15.248%**	2.309	15.205%
Average			**19.522%**		19.078%

Table 5.7: Average RMSE and improvement for fully and locally-connected networks with 24 nodes in the middle layer

Site	Raw RMSE	Locally-connected 24 nodes in the middle layer (Original NN-SDPT)		Fully-connected 24 nodes in the middle layer (Modified NN-SDPT)	
		Average RMSE	Average Improvement	Average RMSE	Average Improvement
1	3.049	2.271	25.520%	2.267	**25.604%**
2	2.858	2.263	20.799%	2.261	**20.857%**
3	2.651	2.154	**18.766%**	2.168	18.214%
4	2.627	2.223	**15.416%**	2.240	14.780%
5	2.564	2.240	12.655%	2.212	**13.741%**
6	2.728	2.028	**25.656%**	2.036	25.339%
7	2.722	2.348	**13.763%**	2.352	13.618%
8	2.594	2.068	**20.280%**	2.072	20.099%
9	2.703	2.060	**23.801%**	2.074	23.257%
10	2.692	1.933	28.175%	1.931	**28.270%**
11	2.645	2.211	**16.367%**	2.221	16.020%
12	2.623	2.089	20.380%	2.088	**20.398%**
13	2.633	2.264	**14.014%**	2.291	12.987%
14	2.652	2.075	**21.736%**	2.096	20.973%
15	2.722	2.312	15.083%	2.307	**15.264%**
Average			**19.494%**		19.295%

Table 5.8: Average improvement for locally-connected networks with 22, 23, 24 and 25 nodes in the middle layer

Site	Locally Connected 25 nodes in middle layer	Locally Connected 24 nodes in middle layer	Locally Connected 23 nodes in middle layer (Original NN-SDPT)	Locally Connected 22 nodes in middle layer
1	25.251%	25.520%	**25.687%**	25.507%
2	21.148%	20.799%	**21.481%**	21.139%
3	18.530%	**18.766%**	18.233%	18.207%
4	14.931%	**15.416%**	14.445%	15.045%
5	12.938%	12.655%	12.516%	**13.098%**
6	26.185%	25.656%	**25.931%**	25.815%
7	**14.243%**	13.763%	14.024%	13.971%
8	20.140%	20.280%	**20.731%**	20.702%
9	24.036%	23.801%	**24.137%**	23.843%
10	28.364%	28.175%	28.630%	**28.885%**
11	16.151%	16.367%	**16.682%**	16.598%
12	20.302%	20.380%	**20.394%**	20.106%
13	13.234%	14.014%	13.890%	**14.106%**
14	21.014%	**21.736%**	20.806%	20.941%
15	15.447%	15.083%	15.248%	**15.587%**
Average	19.461%	19.494%	19.522%	**19.570%**

Table 5.9: Average improvement for the NN-SDPT network and a similar network with one extra layer.

Site	Locally Connected 23 nodes in middle layer (Original NN-SDPT)	Locally Connected 23 nodes in middle layer One extra layer
1	**25.687%**	25.008%
2	**21.481%**	21.269%
3	18.233%	**18.680%**
4	14.445%	**14.924%**
5	**12.516%**	11.478%
6	**25.931%**	24.522%
7	**14.024%**	12.354%
8	**20.731%**	19.762%
9	**24.137%**	23.715%
10	**28.630%**	27.151%
11	**16.682%**	16.247%
12	**20.394%**	19.957%
13	**13.890%**	12.538%
14	**20.806%**	20.123%
15	**15.248%**	14.129%
Average	**19.522%**	18.791%

Table 5.10: Average improvement for the NN-SDPT network similar networks with dropout layers

Site	Locally Connected 23 nodes in middle layer (Original NN-SDPT)	Locally Connected 23 nodes in middle layer One dropout layer (rate=0.1)	Locally Connected 23 nodes in middle layer One dropout layer (rate=0.1)
1	25.687%	**25.761%**	24.262%
2	**21.481%**	21.228%	20.439%
3	**18.233%**	18.031%	16.605%
4	14.445%	**14.513%**	12.890%
5	12.516%	**12.862%**	11.309%
6	**25.931%**	25.663%	24.643%
7	**14.024%**	13.861%	12.273%
8	**20.731%**	19.672%	18.330%
9	**24.137%**	23.568%	22.417%
10	**28.630%**	27.174%	26.609%
11	16.682%	**16.694%**	16.295%
12	**20.394%**	20.278%	18.665%
13	**13.890%**	13.226%	12.323%
14	20.806%	**20.973%**	19.725%
15	15.248%	**15.524%**	14.504%
Average	**19.522%**	19.269%	18.086%

5.7 Summary of this chapter

In this chapter we introduced a model for deterministic machine learning-based correction of wind speed forecasts. The development of this model was inspired by denoising autoencoders: a neural network based architecture designed to eliminate noise from data.

Two networks were introduced for the correction of wind speed: a single-variable input network called NN-S (that receives wind speed only as a input), and a multi-variable network called NN-SDPT (that receives wind speed, direction, temperature and pressure as inputs). Models were evaluated by comparing the results with a baseline method (MOS) and the raw forecasts.

Results show that the denoising autoencoders-based architecture is advantageous at reducing the bias in the NWP models. Both proposed networks were able to output corrected forecasts with reduced RMSE and MAE, and increased CC when compared to the baseline and the raw predictions. More importantly, the multi-variable network NN-SDPT was superior to its single-variable counterpart, showing that the multi-variable neural network was able to find valuable information in other atmospheric variables, and to use that information for the correction of wind speed.

Chapter 6

A deep generative model for probabilistic forecasting

This chapter portrays the second proposed model for machine learning-based correction of wind speed forecasts. In contrast with the model introduced in chapter 5 which was of deterministic nature, the model presented in this chapter is a probabilistic model. We start with a description of its nature and training details. Results are shown in section 6.4, including results of power estimation using power curve data. A discussion about the results is offered at the end of the chapter.

6.1 Description of the model

One of the biggest drawbacks of the deterministic model introduced in chapter 5 was the loss of variance in the model's outputs. This phenomenon is known as the bias-variance trade off [137], and it is a well-known problem when using the MSE as a loss function. Given a collection of m observations $\mathbf{y} = \{y_0, y_1, \ldots, y_m\}$ as their corresponding forecasts $\hat{\mathbf{y}} = \{\hat{y}_0, \hat{y}_1, \ldots, \hat{y}_m\}$, the MSE can be written as follows [138].

$$\text{MSE} = \text{Var}\left(y - \hat{y}\right) + \left[E\left(y\right) - E\left(\hat{y}\right)\right]^2, \tag{6.1}$$

where the first term on the right-hand side of the equation is the variance of the forecast errors, and the second term is a measure of bias. Further decomposition leads to:

$$\mathrm{MSE} = \mathrm{Var}\,(y) + \mathrm{Var}\,(\hat{y}) - 2\mathrm{Cov}\,(y, f) + \left[E\,(y) - E\,(f)\right]^2 . \tag{6.2}$$

From Eq. 6.2 it is clear that minimizing the MSE implies reducing as well the variance in the forecast. If the variance is reduced, the resulting smoothed data will have an increased bias due to the lost detail. On the other hand, attempting to reduce the bias will increase the variance.

As mentioned in chapter 1, understanding the variability of wind is critical to predict potential fluctuations in power generation [30]. Since there is a cubic relation between wind speed and power [29], a small variation in wind speed will result in a large variation in power. This specially true in zones where steep and mountainous terrain can introduce turbulence and cause flow separation [17, 18], such as Japan. Moreover, the relationship between wind speed and wind power contains a large amount of uncertainty and it is highly dependent on site-specific wind dynamics, wind farm layout, topography, etc.

Probabilistic weather models have been highly successful when dealing with this kind of uncertainty. The most straightforward method to obtain probabilistic forecasts is by ensemble forecasting [27]. However, as mentioned in section 1.3.2, the accuracy and reliability of an ensemble forecast is directly proportional to the number of ensemble members, each of which requires a complete run of the NWP model. This translates into a significant computational time. Models able to generate ensemble forecasts from single runs of dynamical models are therefore of interest. The Regression Estimation of Event Probabilities (REEP) [46] method, logistic regression [47, 48], and the analog ensemble method [50] are some examples. However, none of this methods has been validated in areas with remarkably fluctuating wind speed dynamics due to complex topography.

Therefore, in this chapter we introduce a probabilistic model able to generate ensemble forecasts from a single, deterministic run of the NWP model, and we will evaluate it using the data from the wind farm of interest (section 3.1). For this purpose we will employ a generative model as described in section 2.2.2, specifically, the conditional version of the variational autoencoders, which will be described in the next section.

6.1.1 The conditional variational autoencoder

The conditional variational autoencoder (CVAE) was introduced by Sohn et al. [139] as the conditional version of the variational autoencoders proposed by Kingma and Welling [82]. CVAEs allow us to map a single input to multiple possible outputs, i.e. to perform probabilistic inference of a distribution conditioned on deterministic input.

Assuming that y is the variable of interest and vector x is a condition, we are interested in inferring from $p(y|\mathbf{x})$, that is the conditional probability of the y given the condition vector $\mathbf{x}$. If we assume a latent variable model, we can write this distribution as

$$p_\theta\left(y|\mathbf{x}\right) = \int p_\theta\left(y|\mathbf{x},\mathbf{z}\right) p_\theta\left(\mathbf{z}|\mathbf{x}\right) d\mathbf{z}, \tag{6.3}$$

where $\mathbf{z}$ is a vector of n Gaussian latent variables, and the notation p_θ denotes that the distribution p is parameterized by θ. Generally, the integral in Eq. 6.3 is intractable and it is very difficult to find efficient estimators.

A method for efficient estimation of the parameters is stochastic gradient variational Bayes [82]. We introduce a surrogate distribution $q_\phi\left(\mathbf{z}|y,\mathbf{x}\right)$ that approximates the true posterior $p_\theta\left(\mathbf{z}|y,\mathbf{x}\right)$. The parameters ϕ are optimized so that $q_\phi\left(\mathbf{z}|y,\mathbf{x}\right) \approx p_\theta\left(\mathbf{z}|y,\mathbf{x}\right)$.

Given this approximate posterior, we can write the conditional log-likelihood as follows:

$$\log p_\theta\left(y|\mathbf{x}\right) = \mathbb{E}_{q_\phi(\mathbf{z}|y,\mathbf{x})} \log\left(\frac{p_\theta\left(y,\mathbf{x}|\mathbf{z}\right)}{q_\phi\left(\mathbf{z}|y,\mathbf{x}\right)}\right) + \mathbb{E}_{q_\phi(\mathbf{z}|y,\mathbf{x})} \log\left(\frac{q_\phi\left(\mathbf{z}|y,\mathbf{x}\right)}{p_\theta\left(\mathbf{z}|y,\mathbf{x}\right)}\right). \tag{6.4}$$

A complete derivation is given in Kingma and Welling [82] and Sohn et al. [139]. The second term in the right hand side of equation 6.4 corresponds to the Kullback-Leiber (KL) divergence between $p_\theta\left(\mathbf{z}|y,\mathbf{x}\right)$ and $q_\phi\left(\mathbf{z}|y,\mathbf{x}\right)$. Given that the KL divergence is always positive, the first term in the right hand side of Eq. 6.4 becomes a lower bound on the log-likelihood of the data,

$$\log p_\theta\left(y|\mathbf{x}\right) \geq \mathbb{E}_{q_\phi(\mathbf{z}|y,\mathbf{x})}\left[-\log q_\phi\left(\mathbf{z}|y,\mathbf{x}\right) + \log p_\theta\left(y,\mathbf{x}|\mathbf{z}\right)\right]. \tag{6.5}$$

The right hand side of Eq. 6.5 is known as the Evidence Lower Bound (ELBO), so that maximizing the conditional log-likelihood is equivalent to maximizing the ELBO. After some manipulation we can obtain

$$\begin{aligned}\log p_\theta(y|\mathbf{x}) &\geq \mathbb{E}_{q_\phi(\mathbf{z}|y,\mathbf{x})}\left[-\log q_\phi(\mathbf{z}|y,\mathbf{x}) + \log p_\theta(\mathbf{z}|\mathbf{x})\right] + \mathbb{E}_{q_\phi(\mathbf{z}|y,\mathbf{x})}\left[\log p_\theta(y|\mathbf{x},\mathbf{z})\right] \\ &\geq -KL\left(q_\phi(\mathbf{z}|y,\mathbf{x}) \parallel p_\theta(\mathbf{z}|\mathbf{x})\right) + \mathbb{E}_{q_\phi(\mathbf{z}|y,\mathbf{x})}\left[\log p_\theta(y|\mathbf{x},\mathbf{z})\right],\end{aligned} \tag{6.6}$$

where the first term is the KL divergence between the surrogate posterior and the conditional prior (also known as the regularization term), and the second term is the expected log-likelihood of the conditioned data (also known as the reconstruction term). It is clear from Eq. 6.6 that by maximizing the ELBO we are a) minimizing the distance between the surrogate posterior and the prior distribution, and b) maximizing the log-likelihood of $p_\theta(y|\mathbf{x},\mathbf{z})$, resulting in better sample generation [80]. Since the expected log-likelihood can be obtained via Monte Carlo sampling, we can write our final optimization target as

$$\begin{aligned}\mathcal{L}_{\text{CVAE}} &= -\beta KL\left(q_\phi(\mathbf{z}|y,\mathbf{x}) \parallel p_\theta(\mathbf{z}|\mathbf{x})\right) + \frac{1}{L}\sum_{l=1}^{L}\log p_\theta(y|\mathbf{x},\mathbf{z}_l) \\ &= \beta\mathcal{L}_{KL} + \mathcal{L}_R,\end{aligned} \tag{6.7}$$

where L is the number of Monte Carlo samples, and the hyperparameter β is introduced to control the amount of regularization. Note that, in order to implement stochastic gradient-based optimization on Eq. 6.7, the random variables $\mathbf{z}$ must be expressed in a differentiable form. This is done by applying the reparametrization trick [82], where we express $\mathbf{z}$ as a deterministic variable using an auxiliary random variable $\boldsymbol{\epsilon}$, i.e., $\mathbf{z}_l = g_\phi(y,\mathbf{x},\boldsymbol{\epsilon}_l)$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$.

6.1.2 Architecture

Each of the three probability distributions in Eq. 6.7 can be parameterized with neural networks. Here we follow the same naming scheme as in [139], using *recognition network* for $q_\phi(\mathbf{z}|y,\mathbf{x})$, *conditional prior network* for $p_\theta(\mathbf{z}|\mathbf{x})$ and *generation network* for $p_\theta(y|\mathbf{x},\mathbf{z})$. An schematic of the complete model is shown in Figure 6.1.

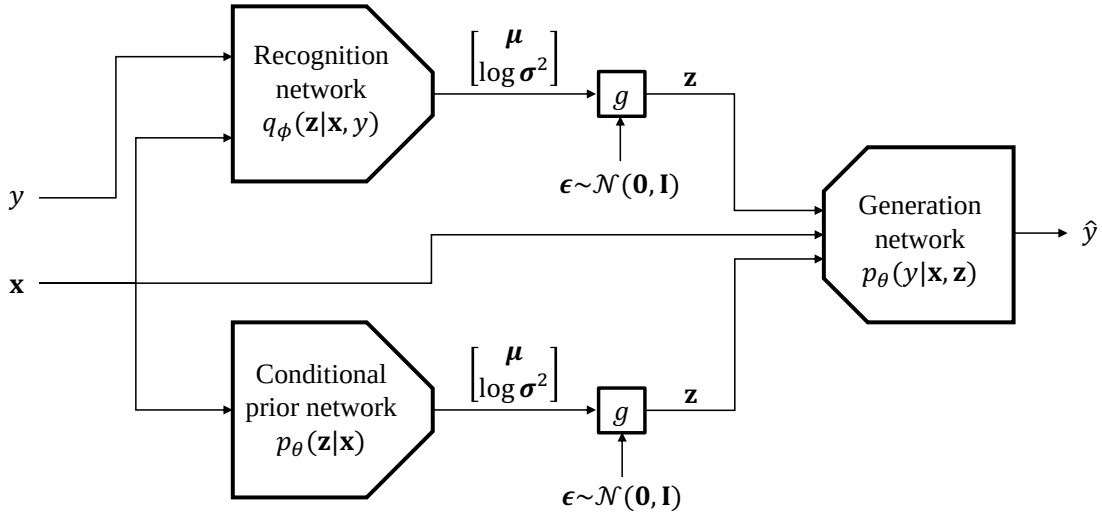

Figure 6.1: Schematic of the conditional variational autoencoder (CVAE). For the recognition and conditional prior networks, the outputs $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ correspond to the mean vector and covariance vector of the distributions each network represents. For the generation network, the output is taken as a direct sampling from the generative distribution.

6.2 Training details

For this model we employ the dataset from the Awaji Island wind farm, as described in chapter 3. Parameter optimization is done by minimizing equation 6.7. During training, latent variables are obtained from both the recognition and the conditional prior networks. The KL term in Eq. 6.7 is calculated based on the outputs from these two networks. The latent variables obtained from the recognition network are used to generate a sample $\hat{y}$, and the reconstruction term in Eq. 6.7 is calculated from y and $\hat{y}$. The recognition network is only used during training; during inference the latent variables from the conditional prior network are used to generate samples. Table 6.1 shows the number of nodes and layer in each of these networks.

For each training instance, the inputs y and $\mathbf{x}$ are the observed wind speed scalar and a prior information vector respectively. The prior information vector $\mathbf{x}$ consists of the forecasted values of wind speed, wind direction, temperature,

Table 6.1: Description of the networks described in Figure 6.1

	Input layer	Hidden layers	Output layer
Recognition network	46	36 26	16
Conditional prior network	45	35 25	16
Generation network	56	35 18	1

pressure and humidity, for each corresponding observation. Wind direction is decomposed into its sine and cosine components. Also, two spatio-temporal features are encoded in the input as well. The corresponding forecasted lead time in hours (0 to 23) and the corresponding turbine number (1 to 15) are included as categorical variables using one-hot encoding. By including these variables we provide relevant information for the recognition of patterns that may occur only at certain sites and/or certain times. Overall, the final size of $\mathbf{x}$ is of $[1 \times 45]$.

Despite our goal being to obtain day-ahead 24-hour forecasts, we have opted for a single-valued 1 hour output and for time information to be included as a one-hot encoding, rather than using a 24-hour input. Preliminary tests showed that the former is able to model correlation between lead times more accurately than the latter, presumably due to overfitting induced by feeding the network 24 time points at the same time.

The data used to train the model is the same as in the previous model, however in this occasion we perform simple validation after training. Therefore, the data for the period from 2016-01-03 to 2020-06-01 was divided into three smaller sets:

1. Training: 2016-01-03 to 2018-12-31 (70% of data)
2. Validation: 2019-01-01 to 2019-12-25 (20% of data)
3. Testing: 2019-12-26 to 2020-06-01 (10% of data)

All results shown correspond to the test set.

Optimization of equation 6.7 was performed by using a cyclical annealing schedule approach [140] for the β hyperparamater in order to avoid the KL-vanishing problem that is common when training CVAE models. ADAM (section 2.3.3) was used as the optimization method and all inputs were normalized before feeding them to the networks (section 2.3.2).

6.3 Baseline method

We have chosen two baseline methods to validate and assess the skill of the presented CVAE approach. First, the MOS model (already introduced in section 5.3) is used as a deterministic baseline method. The MOS parameters were determined by least-square fitting; using the same five meteorological variables used for training the CVAE as predictors and the observations as predictands.

Second, we use the Analog Ensemble (AnEn) as presented by Monache et al. [50] as a probabilistic baseline method. The AnEn is another method of obtaining probabilistic forecasts from a collection of past deterministic model runs.

Given a history of deterministic forecasts and their corresponding observations, an ensemble forecast for time t can be constructed by searching for analogs. In this sense, an analog is defined as a past historical forecast that is similar to the current forecast. The "similarity" or "quality" of the analog is determined by the following metric:

$$\|F_t, A_{t'}\| = \sum_{i=1}^{N_v} \frac{w_i}{\sigma_{f_i}} \sqrt{\sum_{j=-\tilde{t}}^{\tilde{t}} (F_{i,t+j} - A_{i,t'+j})^2}, \tag{6.8}$$

where F_t is the current NWP forecast at time t, $A_{t'}$ is an analog; a past forecast with the same forecast lead time, N_v is the number of variables, w_i are the weights assigned to each variable, σ_{f_i} is the standard deviation of the time series of a given variable used in the search, and $\tilde{t}$ is half the time window used to compute the metric. The observations corresponding to the analogs with the lowest metric are taken as part of the resulting ensemble. Figure 6.2 shows a schematic of the method.

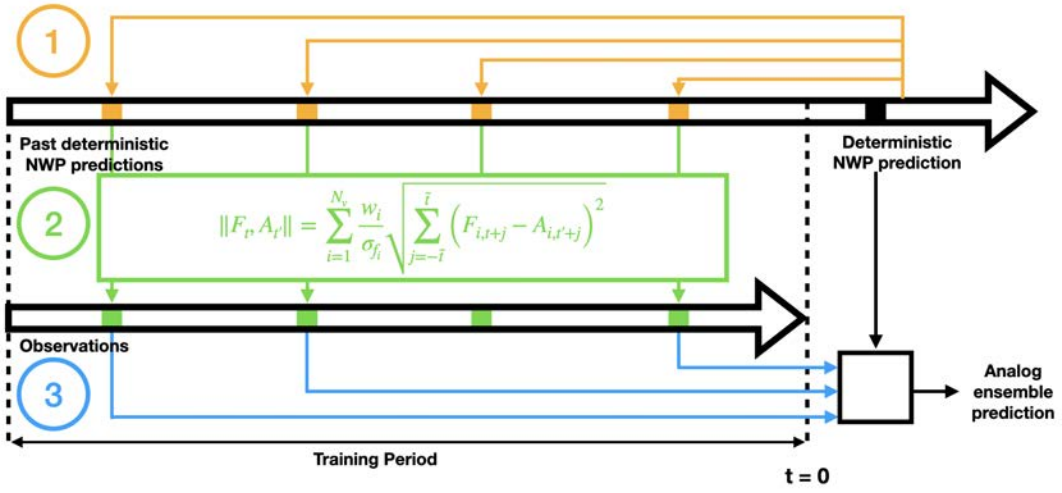

Figure 6.2: Schematic of the analog ensemble method (adapted from [50]). In this example, the resulting ensemble is of size 3.

In order to be fair with the CVAE method, the variables included in the analog search are the same as those included as inputs in the CVAE, and all weights w_i are set to 1. The time window is set to $\tilde{t} = 0.5$ which means that the metric is calculated by using only one time point.

The number of analogs used to construct the analog ensemble is $n_{\text{ens}} = 21$ as in [50] and, therefore, the number of generated forecasts from the CVAE is set to 21 as well. We will refer to the ensemble generated by the CVAE as CVAE-En.

6.4 Results

In this section we show the results of wind speed and wind power forecasts after implementing the CVAE to generate ensemble forecasts. This section is divided into three parts; first we evaluate the CVAE output as results of a deterministic forecast and then we use probabilistic methods tools and metrics to assess its performance as a probabilistic forecast. Finally, we use the results from the previous sections to proceed to wind power prediction stage, where we evaluate the advantages of the CVAE when estimating wind power output from a wind turbine power curve. Also, in this section we use the skill metric (section 3.4) to compare the results of two models.

6.4.1 As a deterministic forecast

First, we evaluate the CVAE results when treated as a deterministic forecast. To perform such a deterministic inference we draw multiple $\mathbf{z}$ from the prior distribution, and use the mean of the generated samples as the prediction.

We start with the results in a per-site basis. The bar chart shown in Figure 6.3 shows the RMSE (left axis) for the uncorrected NWP forecast (referred as "Raw"), the MOS model, and the generated forecast obtained as an average of $n = 21$ samples from the generation network. In the same figure we have included the skill percentage (right axis). The same information is displayed in Table 6.2. It can be seen that the CVAE shows considerably less error and thus higher forecast skill when compared to both the uncorrected NWP and MOS. Similarly, we obtain similar results for the MAE metric shown in Table 6.3 and the CC metric shown in Table 6.4. Here we have also included the delta between skills to highlight that the CVAE can increase the skill up to 13 percentage points.

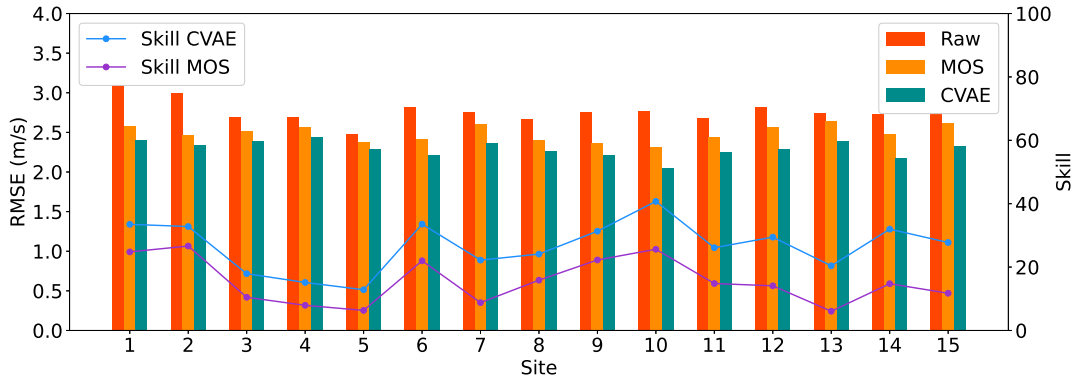

Figure 6.3: Comparison of RMSE and skill values for the NWP output (Raw) and after the correction with MOS and CVAE at each site.

Table 6.2: Root mean square error for the NWP forecast, MOS correction and CVAE generation, for all 15 sites. Δ is the difference of the forecasting skill between CVAE and MOS.

Site	NWP RMSE [m/s]	CVAE RMSE [m/s]	MOS RMSE [m/s]	Skill CVAE [%]	Skill MOS [%]	Δ [%]
1	3.107	2.402	2.585	22.7	16.8	5.9
2	2.997	2.342	2.465	21.9	17.8	4.1
3	2.690	2.388	2.514	11.3	6.6	4.7
4	2.698	2.441	2.563	9.5	5.0	4.5
5	2.475	2.285	2.381	7.7	3.8	3.9
6	2.821	2.210	2.420	21.7	14.2	7.5
7	2.760	2.370	2.606	14.2	5.6	8.6
8	2.669	2.267	2.404	15.1	9.9	5.1
9	2.760	2.209	2.368	20.0	14.2	5.8
10	2.769	2.048	2.316	26.0	16.4	9.7
11	2.687	2.247	2.437	16.4	9.3	7.1
12	2.821	2.285	2.564	19.0	9.1	9.9
13	2.744	2.388	2.638	13.0	3.9	9.1
14	2.728	2.176	2.473	20.2	9.3	10.9
15	2.833	2.325	2.618	18.0	7.6	10.4
Average				16.7	9.5	7.2

Table 6.3: Mean Absolute Error values for the NWP forecast, MOS correction and CVAE generation, for all 15 sites. Δ is the difference of the forecasting skill between CVAE and MOS.

Site	NWP MAE [m/s]	CVAE MAE [m/s]	MOS MAE [m/s]	Skill CVAE [%]	Skill MOS [%]	Δ [%]
1	2.389	1.886	2.098	21.0	12.2	8.8
2	2.274	1.838	1.984	19.3	12.8	6.5
3	2.045	1.743	2.031	7.0	0.6	6.3
4	2.094	1.842	2.045	9.4	2.4	7.0
5	1.925	1.784	1.886	7.3	2.0	5.3
6	2.196	1.662	1.923	22.3	12.4	9.9
7	2.169	1.835	2.069	15.7	4.6	11.1
8	2.094	1.674	1.919	15.6	8.4	7.2
9	2.160	1.629	1.848	21.0	14.5	6.5
10	2.153	1.539	1.814	27.8	15.8	12.0
11	2.059	1.648	1.855	18.2	9.9	8.3
12	2.177	1.721	2.031	19.2	6.7	12.5
13	2.100	1.806	2.042	13.3	2.8	10.5
14	2.109	1.684	1.943	20.9	7.9	13.1
15	2.218	1.796	2.082	18.6	6.1	12.5
Average				17.1	7.9	9.2

Table 6.4: Correlation coefficient values for the NWP forecast, MOS correction and CVAE generation, for all 15 sites. Δ is the difference of the forecasting skill between CVAE and MOS.

Site	NWP CC [m/s]	CVAE CC [m/s]	MOS CC [m/s]	Skill CVAE [%]	Skill MOS [%]	Δ [%]
1	0.624	0.775	0.741	24.3	18.9	5.5
2	0.601	0.751	0.723	25.0	20.4	4.7
3	0.695	0.754	0.727	8.6	4.7	3.9
4	0.706	0.760	0.730	7.5	3.3	4.2
5	0.742	0.784	0.760	5.7	2.5	3.2
6	0.686	0.792	0.742	15.4	8.2	7.2
7	0.706	0.789	0.741	11.7	4.9	6.8
8	0.694	0.762	0.730	9.8	5.2	4.6
9	0.680	0.761	0.719	11.8	5.7	6.1
10	0.670	0.792	0.726	18.3	8.4	9.9
11	0.695	0.763	0.720	9.7	3.5	6.2
12	0.676	0.785	0.719	16.1	6.4	9.7
13	0.705	0.786	0.730	11.4	3.5	7.9
14	0.657	0.772	0.692	17.5	5.4	12.1
15	0.685	0.794	0.729	16.1	6.5	9.5
Average				13.9	7.2	6.8

In a similar fashion, we present in Figure 6.4 the RMSE values for Raw, MOS and the averaged CVAE samples for every of the 24 forecast lead times. The period of strong winds that extends from approximately the 9th forecasted hour (17:00 JST) to the 12th forecasted hour (20:00 JST) (see figures 3.5 and 3.9, section 3.1) translates into a higher RMSE in the raw forecast, as shown by the red line in Figure 6.4. CVAE is able to generate forecasts with reduced error for all lead hours, outperforming MOS as well.

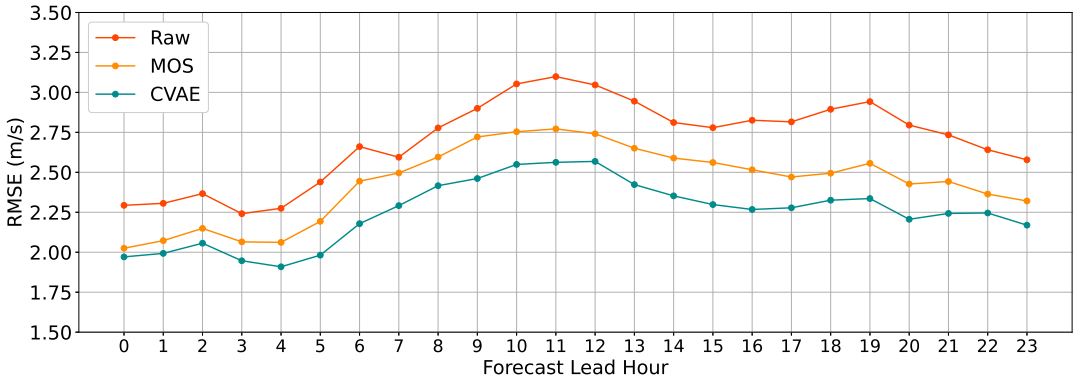

Figure 6.4: Comparison of RMSE and skill values for the NWP output (Raw) and after the correction with MOS and CVAE, for different forecast lead hour.

6.4.2 As a probabilistic forecast

We now focus on evaluating the performance of the CVAE as a generator of probabilistic forecasts. We will be using the diagnostic tools already mentioned in section 3.4.

Table 6.5 shows the CRPS and the RI for the CVAE-En and the AnEn in a per-site basis. It can be seen that the ensemble provided by the CVAE has lower CRPS and scores for all sites. For the aggregated data of all sites, the CVAE-En CRPS value of 1.326 corresponds to a decrease of 18.8% from the corresponding value for AnEn, which is of 1.633. We have also calculated the CRPS and the RI in a hourly basis, as shown in 6.5 (left axis). For all lead times, the CVAE-En shows lower CRPS and RI scores when compared with the AnEn.

Table 6.5: Continuous Ranked Probability Score (CRPS) and Reliability Index (RI) calculated for AnEn and CVAE-En. Δ is the difference between CVAE-En and AnEn.

Site	CRPS CVAE-En	CRPS AnEn	CRPS Δ	RI CVAE-En	RI AnEn	RI Δ
1	1.405	1.858	-0.454	0.413	0.706	-0.294
2	1.380	1.772	-0.393	0.434	0.650	-0.216
3	1.401	1.439	-0.039	0.605	0.582	0.023
4	1.413	1.634	-0.221	0.409	0.573	-0.163
5	1.344	1.494	-0.150	0.412	0.537	-0.125
6	1.290	1.646	-0.355	0.437	0.584	-0.147
7	1.357	1.761	-0.404	0.340	0.675	-0.335
8	1.328	1.554	-0.226	0.456	0.547	-0.090
9	1.287	1.602	-0.315	0.503	0.630	-0.127
10	1.160	1.602	-0.443	0.346	0.632	-0.287
11	1.255	1.549	-0.294	0.361	0.604	-0.242
12	1.322	1.651	-0.329	0.413	0.567	-0.154
13	1.357	1.650	-0.293	0.349	0.535	-0.186
14	1.242	1.573	-0.331	0.342	0.529	-0.187
15	1.346	1.731	-0.385	0.384	0.598	-0.214
All sites	1.326	1.633	-0.307	0.394	0.579	-0.185

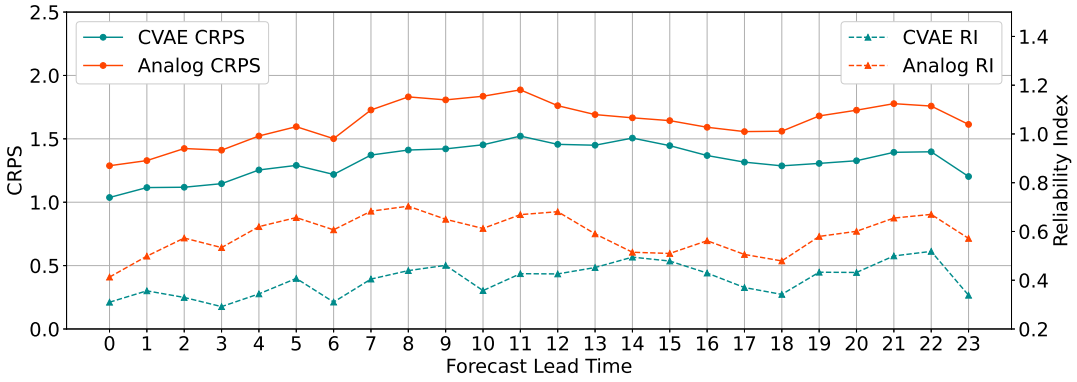

Figure 6.5: Continuous Ranked Probability Score (CRPS, continuous line with circular marker) and Reliability Index (RI, broken line with triangular marker) calculated for AnEn (red) and CVAE-En (green), for each lead time.

Figure 6.6 shows the rank histograms for the aggregated data of all sites and

lead times. As mentioned in section 3.4, the rank histogram of an ideal ensemble would show uniformity, meaning that all ranks would be the same height. The left side panel corresponds to the AnEn, which is not completely symmetric and shows overpopulation at both rank extremes. This means that the AnEn exhibits underdispersion (overconfidence). The right side panel shows the rank histogram for the CVAE-En. This time the rank diagram is more symmetric and centered. It still displays overpopulation at both rank extremes, however with lower relative frequencies when compared to the AnEn.

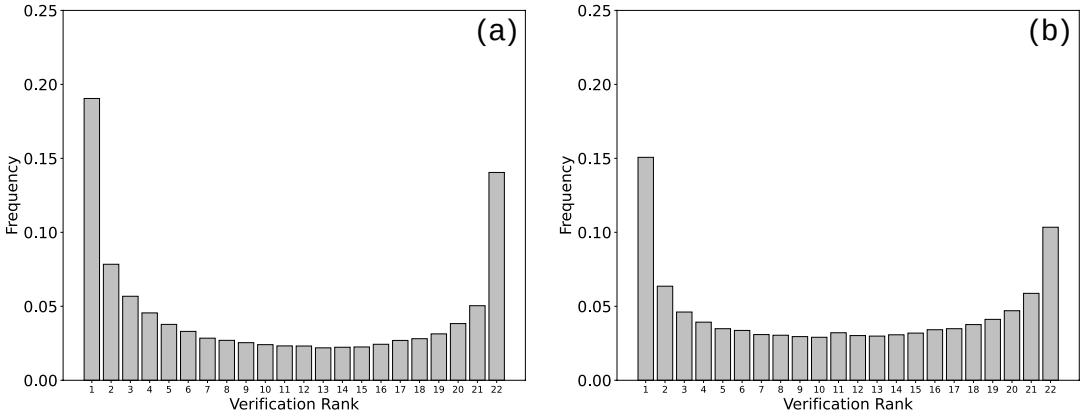

Figure 6.6: Rank Histograms of wind speed forecasts, for (a) AnEn and (b) CVAE-En.

Figure 6.7 shows the dispersion diagrams for the AnEn and CVAE-En. As mentioned in section 3.4, the dispersion diagram of an ideal ensemble would show that the MSE and the variance of the ensemble are equal. In this case, the left side panel corresponding to the AnEn, shows considerable discrepancy with the MSE at all forecast lead times. The right side panel of Figure 6.7 shows the dispersion diagram for the CVAE-En. It reveals that the discrepancy between both values is considerably lower, and consistent along all forecast lead times. Also, the mean square error is lower than that of the AnEn for all forecast lead times.

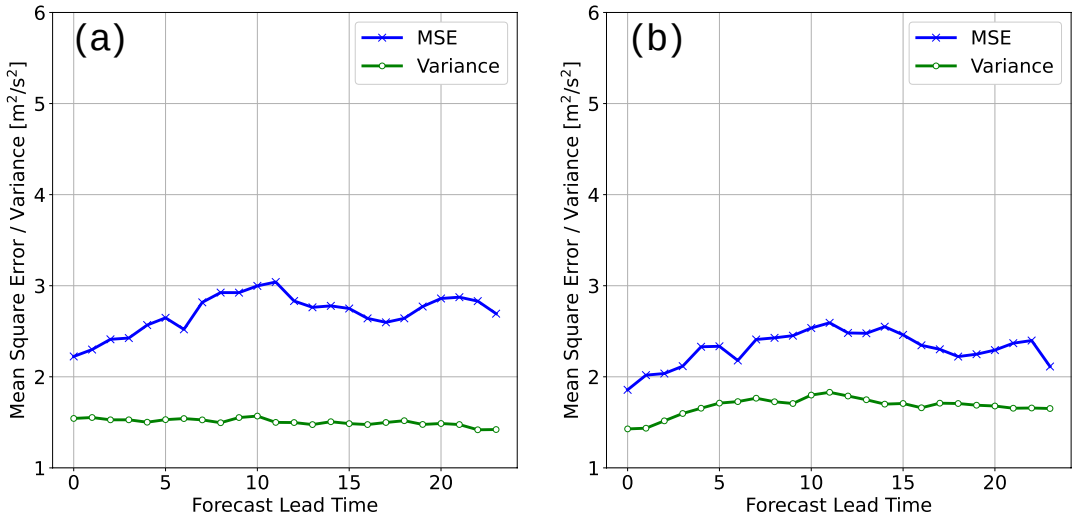

Figure 6.7: Dispersion diagrams of the wind speed forecasts, for (a) AnEn and (b) CVAE-En.

The next set of metrics are the energy and variogram scores, which are calculated by taking a 24-hour prediction as a single multivariable forecast. These shown in Table 6.6 for each of the turbines. The aggregated scores for all sites are shown in the last row. Regarding the energy score, the CVAE-En shows a reduced score for all sites when compared to the AnEn, with a score of 8.009 versus 9.602 for the aggregated data for all sites. For the variogram metric, the CVAE-En scores are lower for all turbines except turbine 3, where the CVAE-En shows a slight increase when compared to the AnEn. Nevertheless, the CVAE-En shows an overall score of 233.579, which is an improvement of over 16 points.

Table 6.6: Energy and Variogram scores calculated for AnEn and CVAE-En. Δ is the difference between CVAE-En and AnEn.

Site	Energy CVAE-En	Energy AnEn	Energy Δ	Variogram CVAE-En	Variogram AnEn	Variogram Δ
1	8.501	10.802	-2.301	259.937	279.615	-19.678
2	8.276	10.310	-2.034	248.932	260.606	-11.673
3	8.460	8.553	-0.093	228.043	225.132	2.912
4	8.545	9.610	-1.065	250.114	261.917	-11.803
5	8.065	8.829	-0.764	235.301	245.445	-10.144
6	7.783	9.695	-1.912	227.367	243.983	-16.616
7	8.270	10.238	-1.968	244.563	264.801	-20.237
8	7.972	9.170	-1.198	233.599	240.233	-6.634
9	7.725	9.428	-1.702	217.819	231.170	-13.351
10	7.068	9.429	-2.361	208.543	234.163	-25.620
11	7.635	9.199	-1.564	217.754	232.064	-14.310
12	7.981	9.751	-1.770	233.645	258.868	-25.223
13	8.207	9.767	-1.560	238.022	256.840	-18.818
14	7.474	9.291	-1.816	214.276	236.214	-21.937
15	8.169	10.099	-1.931	246.837	275.739	-28.902
All sites	8.009	9.602	-1.593	233.579	249.762	-16.184

To finish with the wind speed evaluation, we wish to provide a qualitative analysis of the generated wind speed time series. We have chosen two case studies to graphically demonstrate the strength of the present method. The first case runs from Jan. 26 to Jan. 2 2020 and the second case runs from April 24 to May 1 2019. The time series for the first period is presented in the upper panel of Figure 6.8(a) for CVAE-En (b) for the analog ensemble. For the second period it is presented in and 6.9(a) for CVAE-En and (b) for the analog ensemble. The first case was chosen as it shows relatively steady wind over the period of eight days, while the second case shows low speeds disrupted by sudden bursts that peak approximately at noon every day. In this graph we have also included a 95 % prediction interval for both models.

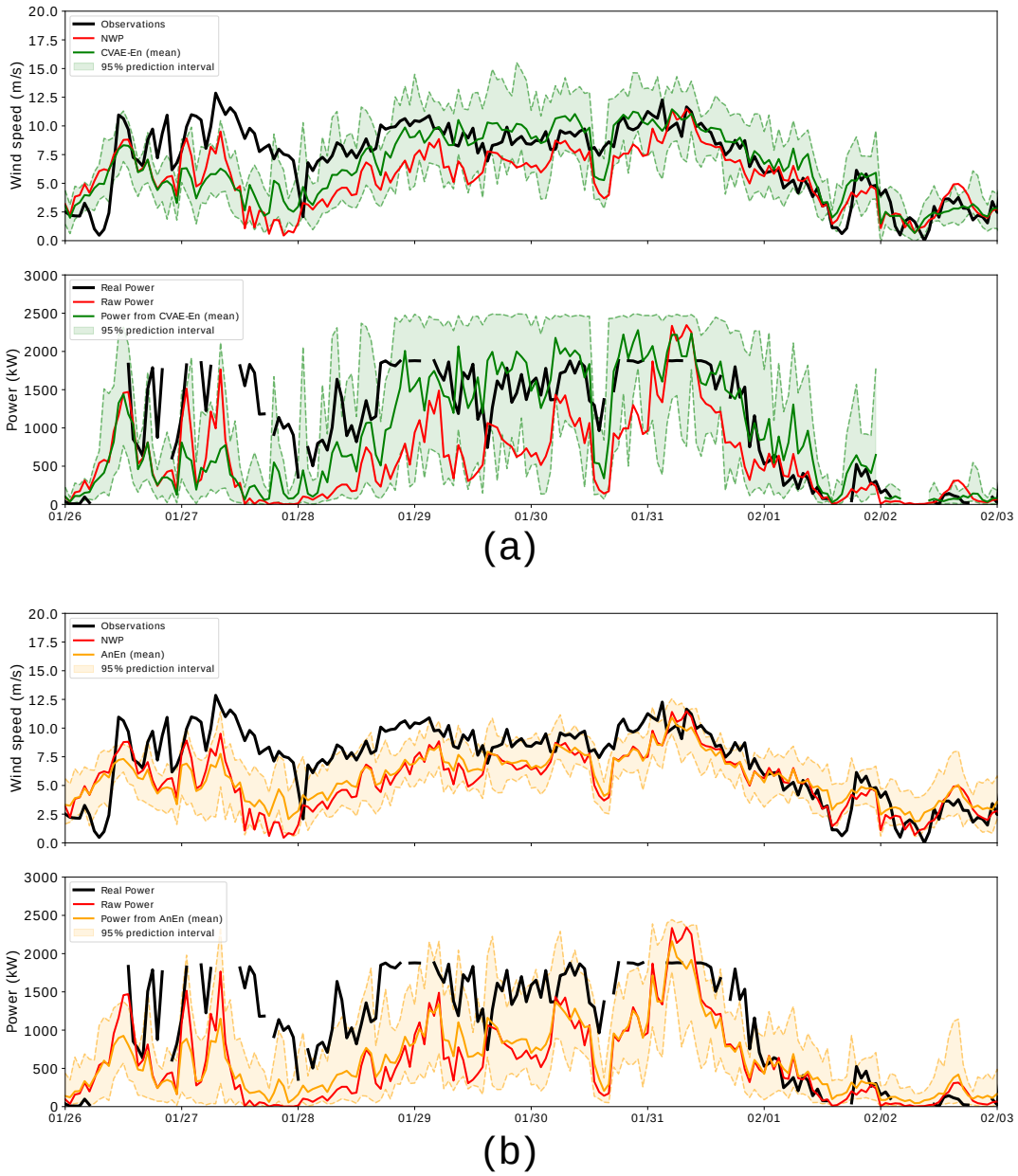

Figure 6.8: Time series (case 1) for (a) CVAE-En and (b) AnEn. All estimations of power were obtained by using the equation in Figure 6.10 and the wind speed predictions out of each model. Raw refers to the power estimation using the NWP wind speed.

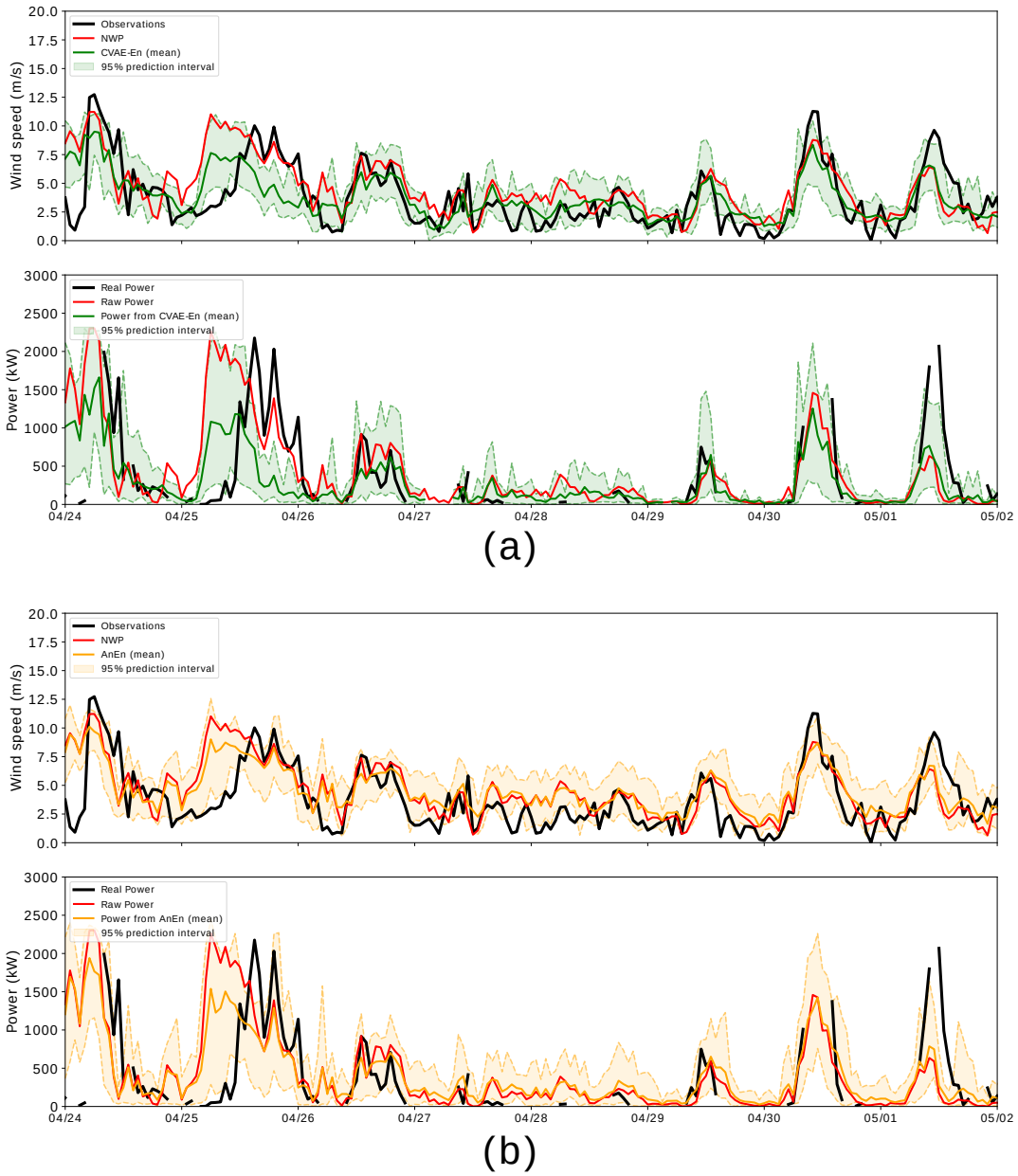

Figure 6.9: Time series (case 2) for (a) CVAE-En and (b) AnEn. All estimations of power were obtained by using the equation in Figure 6.10 and the wind speed predictions out of each model. Raw refers to the power estimation using the NWP wind speed.

Both models are able to provide forecasts that follow the patterns seen in the observations (black thick line). However, it is clear from both figures that the CVAE-En predictions are more centered along the observations than the AnEn. Moreover, an important difference between the CVAE-En and AnEn results is that the CVAE-En shows a 95% prediction interval with variable width. The model

is more certain (thinner interval) about its predictions when the NWP predicted values are low, and it is less certain (thicker interval) when fluctuations are present. On the other hand, the thickness of the interval in the AnEn seems to be largely constant, without any significant variation.

6.5 Power estimation

The remaining of this section is dedicated to the estimation of wind power by using the forecasts generated by the CVAE and the non-linear power curve method specified in section 3.12. We show results comparing predicted and actual wind power for one of the 15 turbines in the wind farm of interest.

Similarly to the previous section, we judge the skill of the power estimations with both deterministic and probabilistic scores. We use the wind speed ensemble members in our non-linear power curve model to obtain a wind power ensemble, and we present the results using scalar metrics and graphical tools.

Figure 6.10 shows the power curve used in this section. Data corresponds to turbine No. 2, collected from 2019/6/1 to 2020/7/1. The non-linear model previously shown in equation 3.1 was fitted to the data, with final parameters $p_{\max} = 2494$, $\alpha = 13.66$, $\beta = 10.66$ and $k = 0.197$.

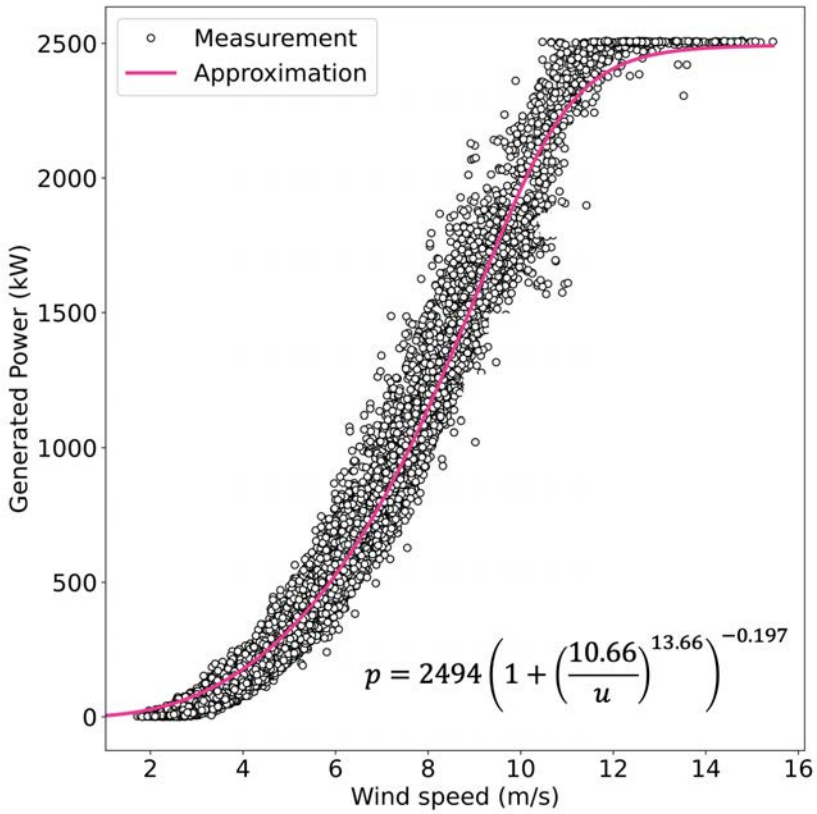

Figure 6.10: Power curve for turbine No.2, using data collected from 2019/6/1 to 2020/7/1. Outliers were removed by using Tukey's method (section 3.3.1)

We start with the scalar metrics shown in table 6.7: RMSE, MAE, CRPS and reliability index for the predicted wind power. The CRPS, while mostly used to evaluate weather forecast, can be used as a measure of performance for any type of probabilistic prediction model [111]. For all metrics, the CVAE-En results in better scores than the AnEn.

Table 6.7: RMSE, MAE, CRPS and reliability index for the wind power probabilistic predictions, for the whole testing period. RMSE and MAE were calculated by using the mean of the ensemble predictions.

Metric	CVAE-En	AnEn
RMSE	605.45	722.14
MAE	438.41	531.72
CRPS	330.05	412.57
RI	0.42	0.61

Figure 6.11 shows the rank histograms for the calculated wind power predictions. A similar behavior to the wind speed rank histograms (Figure 6.6) can be seen, the CVAE-En shows less overpopulation at the extreme ranks and more uniformity. This is confirmed by the reliability index shown in Table 6.7, where CVAE-En has a score of 0.42 while that of AnEn is 0.61. Additionally, the dispersion diagram for the wind power predictions is shown in Figure 6.12. Similarly to wind speed (Figure 6.7), the CVAE-En plot has a smaller MSE and also a smaller discrepancy between the MSE and variance, when compared to the AnEn.

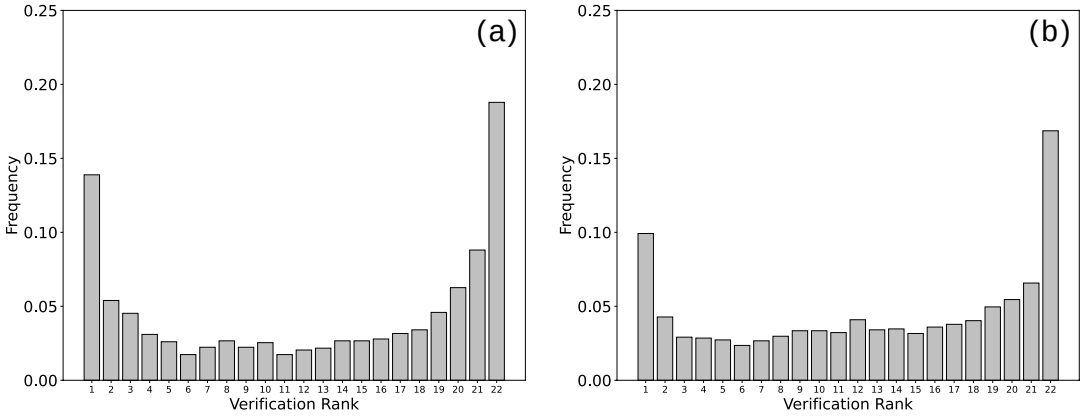

Figure 6.11: Rank Histograms of the wind power predictions, for (a) AnEn and (b) CVAE-En.

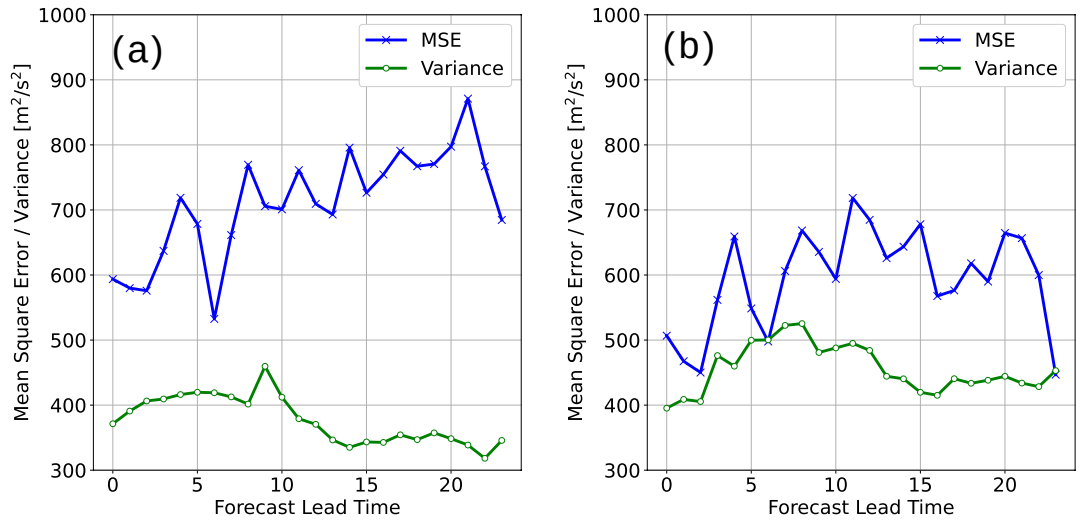

Figure 6.12: Dispersion diagrams of the wind power predictions, for (a) AnEn and (b) CVAE-En.

Finally, we can have a look into the predicted wind power time series by going back once more to figures 6.8 and 6.9. Wind power is plotted directly underneath of its respective wind speed time series, where (a) shows the results for AnEn and (b) the results for CVAE-En. In the first case (figure 6.8) where strong wind speed cause power to go up to $p_{\max}$, the CVAE-En offers a much better estimation than the AnEn. The second case (figure 6.9) show the opposite scenario, in which weak wind speed causes almost no power generation. In this case the CVAE-En shows smaller prediction intervals closer to zero when compared to AnEn.

6.6 Discussion

With respect to the evaluation as a deterministic approach, both Figures 6.3 and 6.4 clearly show that the CVAE is able to generate forecasts with reduced error for all sites and all hours. Moreover, the model is able to recognize the spatio-temporal information that is included in the inputs in the form of one-hot encodings. This approach yielded better results than training individual models for each site and/or hour.

Regarding the probabilistic metrics, it is clear that the low CRPS scores suggest that the CVAE-En results are more likely to be centered around the observed values. The rank histograms are difficult to visually evaluate since both AnEn and CVAE-En show similar profiles although, as mentioned before, the CVAE-En histogram looks slightly more uniform with less overpopulation at the extremes. The RI metric shown in figure 6.5 and table 6.5 helps quantify this difference since, as mentioned in 3.4 this metric is related to the rank histogram. Since the CVAE-En shows a lower RI score for all sites and all hours, it is safe to assume that overall the CVAE-En are slightly more statistically uniform than the AnEn results.

Nevertheless, we cannot say that any of the panels in Figure 6.6 is indeed a uniform ensemble. We claim that this is due to the highly fluctuating nature of the wind in the area of study. Since the NWP is not able to originally reproduce outliers in the wind occurrences, we can argue that searching for past analog NWP forecasts is unreliable in zones with high wind speed fluctuations.

The same argument can be applied for the dispersion diagrams in figure 6.7. While none of the diagrams allow us to say that the ensembles are statistically consistent, there is a clear improvement with the CVAE-En results. A lower MSE error and higher variance are both very desirable qualities in the context of wind speed prediction.

The time series plots confirm once again what we saw with the rank histograms. The AnEn misses many of the observations; in the upper panel of Figure 6.8 (a) the observations are above of the prediction interval at several times such as 01/29 and 01/31, while in the upper panel of Figure 6.9 (a) the observations lie below the prediction interval, such as in 04/28.

We should once again remark the variable width of the CVAE-En prediction interval. One of the motivations to employ probabilistic models is to obtain information about the uncertainty of the prediction. In a reliable probabilistic model it is to expect less uncertainty when the wind is weak and vice versa, however we are not able to see this pattern in the AnEn results. Therefore, the prediction interval for the AnEn provides very little uncertainty information, while the CVAE-En is able to provide it.

It should be mentioned that the analog ensemble method has been shown to produce uniform ensembles in other wind prediction tasks such as in Monache et al. [50]. However, in this study it has produced underdispersed results that tend to miss many of the outliers. As mentioned before, we argue that this is due to the nature of the wind in the study area and it evidences the difficulty of accurately predicting wind speed in a zone with complex geographic features.

Finally, regarding the power estimation shown in section 6.5, we can make two general comments. First, the skills of the models in predicting wind speed accurately translates directly into a good performance when predicting wind power. The CVAE-En wind power time series shows therefore much reliable results. The rank histogram and dispersion diagram for the CVAE-En are superior than those for the AnEn, while in the time series plots at most time points the real power is within the prediction interval. Second, it can be seen that the prediction intervals in figures 6.8 and 6.9 are much thicker for the power estimation than for the speed estimation. This can be attributed to the cubic relationship between wind

speed and power, as any small variation in wind speed will translate into a large variation in power. This also highlights the need to predict wind speed accurately, since any small error in speed will cause large errors in power.

Compared to the pure-based wind speed prediction model in chapter 4 and the deterministic correction method in chapter 5, the model presented in this section has shown the most advantages. Similarly to the NN-S and NN-SDPT networks and contrary to the VMD+LSTM approach, the CVAE in this chapter does not show decrease of performance at longer prediction times. On the other hand, the uncertainty information provided by the probabilistic sampling is of high value for the task of power estimation. The fact that a probabilistic ensemble forecast can be obtained from a single NWP run is of enormous value. Therefore, we can argue that the model in this chapter is suitable and adequate for wind speed and wind power estimations.

6.7 Summary of this chapter

In this chapter we introduced the second proposed model for the correction of wind speed forecasts. This model is of probabilistic nature. The conditional variational auto-encoder (CVAE) approach was used to generate enhanced ensemble forecasts from deterministic NWP predictions. Both deterministic and probabilistic metrics were used to evaluate the model. Two baseline methods were employed: the MOS method for deterministic evaluation, and the analog ensemble method for probabilistic evaluation.

Results show that the CVAE is able to provide ensembles with superior properties when compared to both MOS and the analog ensemble. From a deterministic point of view, the CVAE results presented reduced RMSE and MAE. From a probabilistic point of view, the CVAE ensemble presented reduced CRPS and RI. There was also improvement in the statistical consistency and uniformity tests, shown in the rank histograms and dispersion diagrams. However, this improvement was minimal.

Finally, a power curve model was constructed from past speed and power data, and used to obtain power estimations with the CVAE results. It was seen that

wind speed predictions with reduced error result directly in power estimations with reduced error as well. The CVAE power estimations were, therefore, superior to those of the analog ensemble.

Chapter 7

Conclusions and future work

This last chapter provides an overall summary of this work, some comments and the final conclusions. Before closing, we mention different directions that can be present thesis can take as future work.

7.1 Summary of this thesis

To reverse and mitigate the effects of climate change and limit global warming below 2°C it is necessary to transition to renewable energy generation. Four technologies lead this transition: hydropower, wind power, photovoltaic solar and biofuels, and several countries are already able to generate more than 20% their energy needs from RES. However the case of Japan is different, in the Asian country the transition has been slow when compared to other nations. This is despite their considerable potential for renewable energy generation, including wind power.

Resource availability is not sufficient. To guarantee grid connectivity and reliability when there are variable renewable resources (such as solar and wind) it is necessary to obtain power output estimations. This implies the need for accurate weather forecasting. In the case of Japan, obtaining wind forecasts is complicated by the complex mountainous terrain of the country. A literature review revealed that methods for improving weather forecasts targeted at power estimation in zones with rugged terrain is virtually non-existing.

The main objective of this thesis was therefore stated in section 1.5. It is of interest to develop wind speed predictions models focused at obtaining wind power

estimations. Machine learning approaches were chosen due to their versatility and the availability of both observed and predicted data. This data comes from a wind farm located in a zone with mountainous and steep terrain, therefore providing the model information about complex and turbulent wind profiles. The machine learning models and methods were introduced in chapter 2. Neural networks served as basis for all of the developed models, which included long short-term memory networks, and generative models. Three approaches were explored: a pure data-based wind speed prediction model, a deterministic NWP forecast correction model and a probabilistic NWP forecast correction model.

The observations and predictions, power model and evaluation tools employed were described in chapter 3. Wind speed data was obtained from two wind farms located in Awaji Island and the Houhoku region, which are zones with rugged terrain and complex landform that result in highly fluctuating wind speed patterns. Measured wind speed data was obtained for each of turbine in the wind farms by using anemometers positioned on top of the nacelle after the rotating blades. For the Awaji Island site, numerical weather predictions were obtained by using the NWP model with initial and boundary conditions from GFS and a 4-domain grid implementation. With respect of the power estimation model, a non-linear power-curve model was chosen to fit wind and power data.

The first model consisted of a pure data-based model for wind speed prediction and it was introduced in chapter 4. The approach consisted of treating past wind speed observations as amplitude-modulated-frequency-modulated (AM-FM) signals in order to apply variational mode decomposition (VMD). The resulting sub-signals were forecasted ahead in time by using long short-term memory (LSTM) networks to obtain 24-hour day ahead forecasts. Results showed that for the present dataset and the employed VMD implementation, decomposing wind speed time series did not resulted in pure harmonics that would have been easier to predict. More importantly, the reconstructed predictions showed satisfactory results for the immediate prediction times, but their performance decrease drastically with time. Both root mean square error (RMSE) and the mean absolute error (MAE) increased sharply after a few time steps, from 1.36 m/s to 3.02 m/s in the case of the RMSE, and from 1.02 to 2.37 m/s in the case of the MAE. The

correlation coefficient (CC) showed also a drastic decline, from 0.87 to practically zero. We finalized this chapter by concluding that pure data-based models are inadequate for wind speed forecasting and that the inclusion of NWP models is necessary.

Chapter 5 introduced a deterministic model for machine learning-based correction of wind speed forecasts. Development was inspired by denoising autoencoders, which are a special neural network architecture designed to eliminate noise from data. Two networks were introduced: a single-variable input network (NN-S, for wind speed) and a multi-variable input network (NN-SDPT for wind speed, wind direction, pressure and temperature). Results showed that both models were able to reduce the bias in the NWP results for all sites and all forecast lead times. Also, both models were superior when compared to the Model Output Statistics (MOS) baseline methods. On average, the NN-S and NN-SDPT networks, respectively, reduced the root mean square error by 11.55% and 16.52% and the mean absolute error by 10.74% and 16.32%, and improved the correlation coefficient by 8.21% and 13.20%. As a matter of fact, the multi-variable network was superior in most metrics when compared to the single-variable network. Also, the multivariable network was able to recover some of the variance lost by the single-variable network. We wrapped up this chapter by concluding that hybrid NWP and data-based methods are much more adequate for wind speed prediction (compared to the pure data-based method in the previous chapter). We also conclude that there is hidden, valuable information about the state of the atmosphere in other variables such as wind direction, pressure and temperature forecasts, and including these forecasts enhances the wind speed correction process.

Finally, chapter 6 introduced a probabilistic model for the correction of wind speed forecasts. This model was based on the conditional variational autoencoder (CVAE) generative approach to obtain enhanced ensemble forecasts from deterministic NWP predictions. The input of the model (the condition) consisted of a multivariable forecast (similar to the NN-SDPT network), but also contained spatio-temporal features. Ensemble forecasts were obtained by sampling repeatedly from the generation network for a given input. The model was evaluated from both a deterministic perspective (by using the mean of the ensemble) and

from a probabilistic perspective (by using the ensemble). Results showed that the CVAE was able to provide ensembles with superior properties when compared the MOS and analog ensemble baseline methods. From a deterministic point of view, the CVAE was able to, on average, reduce the root mean square error by 16.7% and the mean absolute error by 17.1%, and improve the correlation coefficient by 13.9%. From a probabilistic point of view, the CVAE ensemble showed better probabilistic skills and slightly improved rank histograms and dispersion diagrams when compared to the analog ensemble. A power curve model was constructed from past speed and power data and used to obtain power estimations with the CVAE results. It was seen that wind speed predictions with reduced error result directly in power estimations with reduced error as well. This chapter was concluded with the argument that probabilistic correction methods offer the best alternative for wind speed and power estimation, when compared to the methods in previous chapters.

7.2 Conclusions

After consideration of the results shown so far, we can conclude the following points:

- Machine learning methods based on neural networks can successfully be applied for wind speed correction and provide useful power estimations that can aid in the integration of wind power into the electric grid.
- Hybrid approaches that combine both past observations and forecasts from a numerical weather prediction model, are vastly superior to pure data-based approaches.
- Machine learning based methods can be useful in locations with complex geography where wind profiles are diverse and can fluctuate rapidly.
- Accurate estimates of wind speed from these models can be readily used to obtain short term estimates of wind power that allow for day-ahead scheduling operations.

Overall, we were able to obtain models targeted at wind power estimation in zones of rugged terrain. Therefore we can conclude that the main goal of this thesis stated in section 1.5 was accomplished successfully.

7.3 Final comments

Firstly, it should be mentioned that, despite being one of the most important variables that affect turbine output power, wind direction has been largely ignored in this work. Wind direction forecasts were obtained from the NWP model and used as inputs for the correction models, however, the lack of actual wind direction measurements did not allow us to fully incorporate this variable into our studies. It should be stressed that, by nature, wind is a vector quantity and a proper understanding of the direction is critical to obtain more accurate and richer power estimates.

With respect to the pure data-based model presented in chapter 4, we argue it demonstrated how statistical methods are unable to provide reliable forecasts for day-ahead estimation. This implies that the atmospheric physics behind the process are still dominant when compared to the stochastic component. The models presented in the subsequent chapters 5 and 6 are more stable in terms of time, and therefore their use is more adequate for this thesis purpose. It is clear that NWP models provide highly valuable information that is undeniably necessary for accurate forecasting.

The deterministic and probabilistic models introduced were indeed able to enhance the results of the NWP model. Two significant outcomes need to be highlighted. First, the inclusion of additional information about the state of the atmosphere aids in the correction of only one variable. Specially in the case of network NN-SDPT, it could be seen that including the extra variables as input resulted in a wind forecast with higher standard deviation, closer to the standard deviation of the observations (figures 5.8 and 5.9). This suggests that one of the biggest drawbacks of deterministic models, i.e. the reduced variance in the solution, can be mitigated by the addition of supplementary variables.

Second, both deterministic and probabilistic models show very similar improve-

ment percentages for the metrics RMSE, MAE and CC, with the probabilistic model being slightly superior. However, it should be mentioned that the uncertainty information provided by the CVAE model is of very high value, specially when taking into account the time it takes to obtain the generated ensembles. Compared to running real NWP ensembles, having them generated by a machine learning model can save considerable time and computational resources, specially in time sensitive applications such as power unit scheduling.

As a final comment we should remark that although the models shown here were trained with strong and fluctuating winds in mind, the same models can be implemented anywhere observed and predicted data are available.

7.4 Future work

In this last section we indicate some new directions worthy of future research. Firstly, the most apparent gap in the current work is the lack of wind direction measurements that compliment the wind speed measurements. As a first step to improve the presented models, wind direction measurements have to be included preferably combined with wind speed in the form of vector quantities. The multivariable design of the models allow for extra inputs to be taken into account, therefore, inclusion of decomposed x and y wind components is straightforward. In the case of the probabilistic model, new formulations can also be derived that adjust to the calculation of corrected wind direction. Moreover, real wind direction data would also help understand the behavior of wind by hour of the day, and provide insights into the pattern seen in figure 3.5.

As shown in section 5.5.1, it was possible to have a look at the weights of the deterministic multi-variable network NN-SDPT and obtain information about the contribution of each variable to the correction process. The output of the models had a significant component derived from the temperature data in the input. We consider this deserves further research. Neural networks or other machine learning methods can indeed be employed to find meaningful relationships in data that are not evident visually. This ability can be used to find better predictors, and improve models even further.

We also consider that the advantages of the CVAE and other deep learning generative methods deserves further exploration. Methods to integrate power data into the enhancement process must be investigated. Also, approaches to include uncertainty information from the initial and boundary conditions, topography information and additional spatio-temporal correlations into the model are necessary.

The end goal, is after all, to obtain power forecasts that are helpful for RES grid integration and power management. For this reason it is also necessary to quantify the value of the models' forecasts in the decision-making process of choosing one energy source over another. This is known as *forecast value assessment* [73]. This can be achieved by running case studies that include real information from the electricity market: energy retailers, generation companies, electricity price, etc. If a model yields satisfactory results in these case studies, the next step would be test in in power system applications: bidding on electricity markets, virtual power plants, microgrids, unit scheduling and commitment, etc.

Bibliography

[1] United Nations General Assembly, "Resolution A/RES/70/1: Transforming our world: the 2030 Agenda for Sustainable Development," October 2015.

[2] Intergovernmental Panel on Climate Change (IPCC), "Climate Change 2022: Mitigation of Climate Change," 2022.

[3] M. Z. Jacobson, M. A. Delucchi, Z. A. Bauer, S. C. Goodman, W. E. Chapman, M. A. Cameron, C. Bozonnat, L. Chobadi, H. A. Clonts, P. Enevoldsen, J. R. Erwin, S. N. Fobi, O. K. Goldstrom, E. M. Hennessy, J. Liu, J. Lo, C. B. Meyer, S. B. Morris, K. R. Moy, P. L. O'Neill, I. Petkov, S. Redfern, R. Schucker, M. A. Sontag, J. Wang, E. Weiner, and A. S. Yachanin, "100% Clean and Renewable Wind, Water, and Sunlight All-Sector Energy Roadmaps for 139 Countries of the World," *Joule*, vol. 1, no. 1, pp. 108–121, 2017.

[4] M. R. Hannah Ritchie and P. Rosado, "Energy," *Our World in Data*, 2020. https://ourworldindata.org/energy.

[5] International Energy Agency, "Fuels and technologies."

[6] U.S. Energy Information Administration, "Biofuels explained."

[7] International Energy Agency, "Electricity information," tech. rep., International Energy Agency, 2020.

[8] A. Q. Al-Shetwi, M. Hannan, K. P. Jern, M. Mansur, and T. Mahlia, "Grid-connected renewable energy sources: Review of the recent integration requirements and control methods," *Journal of Cleaner Production*, vol. 253, p. 119831, 2020.

[9] S. Y. Abujarad, M. Mustafa, and J. Jamian, "Recent approaches of unit commitment in the presence of intermittent renewable energy resources: A review," *Renewable and Sustainable Energy Reviews*, vol. 70, pp. 215–223, 2017.

[10] J. Cochran, M. Miller, O. Zinaman, M. Milligan, D. Arent, B. Palmintier, M. O'Malley, S. Mueller, E. Lannoye, A. Tuohy, B. Kujala, M. Sommer, H. Holttinen, J. Kiviluoma, and S. K. Soonee, "Flexibility in 21st century power systems," 5 2014.

[11] W. M. Chen, H. Kim, and H. Yamaguchi, "Renewable energy in eastern Asia: Renewable energy policy review and comparative SWOT analysis for promoting renewable energy in Japan, South Korea, and Taiwan," *Energy Policy*, vol. 74, pp. 319–329, 2014.

[12] Y. Che, *Development of an integrated system for wind power forecasting under complex geographic conditions*. PhD thesis, Tokyo Institute of Technology, 2017.

[13] Ministry of Economy, Trade and Industry, "Japan's 2050 carbon neutral goal."

[14] Ministry of Environment, "Study of potential for the introduction of renewable energy," tech. rep., 2011.

[15] The International Renewable Energy Agency, "Renewable Energy Statistics 2020," tech. rep., The International Renewable Energy Agency, Abu Dhabi, 2020.

[16] National Weather Service, "Origin of wind."

[17] X. Han, D. Liu, C. Xu, and W. Z. Shen, "Atmospheric stability and topography effects on wind turbine performance and wake properties in complex terrain," *Renewable Energy*, vol. 126, no. April, pp. 640–651, 2018.

[18] E. S. Politis, J. Prospathopoulos, D. Cabezon, K. S. Hansen, P. K. Chaviaropoulos, and R. J. Barthelmie, "Modeling wake effects in large wind

farms in complex terrain: the problem, the methods and the issues," *Wind Energy*, vol. 15, pp. 161–182, jan 2012.

[19] D. S. Wilks, *Statistical Methods in the Atmospheric Sciences*. Academic Press, 3rd ed., 2011.

[20] S. Rodrigues Moreno, R. Gomes da Silva, V. Cocco Mariani, and L. dos Santos Coelho, "Multi-step wind speed forecasting based on hybrid multi-stage decomposition model and long short-term memory neural network," *Energy Conversion and Management*, vol. 213, p. 112869, 2020.

[21] M. A. Ghorbani, R. Khatibi, M. Fazelifard, L. Naghipour, and O. Makarynskyy, "Short-term wind speed predictions with machine learning techniques," *Meteorology and Atmospheric Physics*, vol. 128, 09 2015.

[22] C.-J. Huang and P.-H. Kuo, "A short-term wind speed forecasting model by using artificial neural networks with stochastic optimization for renewable energy systems," *Energies*, vol. 11, no. 10, 2018.

[23] A. V. Jackson, "Handbook of atmospheric science: principles and applications," 2003.

[24] L. Zjavka, "Wind speed forecast correction models using polynomial neural networks," *Renewable Energy*, vol. 83, pp. 998–1006, 2015.

[25] S. Vannitsem, "Dynamical properties of MOS forecasts: Analysis of the ECMWF operational forecasting system," *Weather and Forecasting*, vol. 23, pp. 1032–1043, 2008.

[26] H. Liu, H.-Q. Tian, C. Chen, and Y.-f. Li, "A hybrid statistical method to predict wind speed and wind power," *Renewable Energy*, vol. 35, pp. 1857–1861, 2010.

[27] C. E. Leith, "Theoretical skill of monte carlo forecasts," *Monthly Weather Review*, 1974.

[28] C. Nayar, S. Islam, H. Dehbonei, K. Tan, and H. Sharma, "28 - power electronics for renewable energy sources," in *Power Electronics Handbook*

(Third Edition) (M. H. Rashid, ed.), pp. 723–766, Boston: Butterworth-Heinemann, third edition ed., 2011.

[29] A. Betz, *Introduction to the Theory of Flow Machines*. Pergamon, 1966.

[30] M. Lydia, S. S. Kumar, A. I. Selvakumar, and G. E. Prem Kumar, "A comprehensive review on wind turbine power curve modeling techniques," *Renewable and Sustainable Energy Reviews*, vol. 30, pp. 452–460, 2014.

[31] M. Zamani, G. H. Riahy, and A. J. Ardakani, "Modifying power curve of variable speed wind turbines by performance evaluation of pitch-angle and rotor speed controllers," *2007 IEEE Canada Electrical Power Conference*, pp. 347–352, 2007.

[32] M. Lydia, A. I. Selvakumar, S. S. Kumar, and G. E. P. Kumar, "Advanced algorithms for wind turbine power curve modeling," *IEEE Transactions on Sustainable Energy*, vol. 4, pp. 827–835, 2013.

[33] V. Thapar, G. Agnihotri, and V. K. Sethi, "Critical analysis of methods for mathematical modelling of wind turbines," *Renewable Energy*, vol. 36, no. 11, pp. 3166–3177, 2011.

[34] M. G. Khalfallah and A. M. Koliub, "Suggestions for improving wind turbines power curves," *Desalination*, vol. 209, no. 1, pp. 221–229, 2007. The Ninth Arab International Conference on Solar Energy (AICSE-9), Kingdom of Bahrain.

[35] C. Carrillo, A. Obando Montaño, J. Cidrás, and E. Díaz-Dorado, "Review of power curve modelling for wind turbines," *Renewable and Sustainable Energy Reviews*, vol. 21, pp. 572–581, 2013.

[36] M. Jafarian and A. Ranjbar, "Fuzzy modeling techniques and artificial neural networks to estimate annual energy output of a wind turbine," *Renewable Energy*, vol. 35, no. 9, pp. 2008–2014, 2010.

[37] S. Akdağ and Ö. Güler, "A comparison of wind turbine power curve models," *Energy Sources*, vol. Part A: Recovery, pp. 2257–2263, 10 2011.

[38] J. Gottschall and J. Peinke, "How to improve the estimation of power curves for wind turbines," *Environmental Research Letters*, vol. 3, p. 015005, jan 2008.

[39] T. Jin and Z. Tian, "Uncertainty analysis for wind energy production with dynamic power curves," *2010 IEEE 11th International Conference on Probabilistic Methods Applied to Power Systems*, pp. 745–750, 2010.

[40] A. Kusiak, H. Zheng, and Z. Song, "On-line monitoring of power curves," *Renewable Energy*, vol. 34, no. 6, pp. 1487–1493, 2009.

[41] A. Kusiak, H. Zheng, and Z. Song, "Models for monitoring wind farm power," *Renewable Energy*, vol. 34, no. 3, pp. 583–590, 2009.

[42] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Fluids Engineering, Transactions of the ASME*, vol. 82, pp. 35–45, 1960.

[43] H. R. Glahn and D. A. Lowry, "The Use of Model Output Statistics (MOS) in Objective Weather Forecasting," 1972.

[44] T. Gneiting, A. E. Raftery, A. H. Westveld, and T. Goldman, "Calibrated Probabilistic Forecasting Using Ensemble Model Output Statistics and Minimum CRPS Estimation," *Monthly Weather Review*, vol. 133, pp. 1098–1118, 2005.

[45] J. B. Bremnes, "Probabilistic Forecasts of Precipitation in Terms of Quantiles using NWP Model Output," *Monthly Weather Review*, vol. 132, pp. 338–347, 2004.

[46] H. Glahn, *Statistical weather forecasting*, pp. 289–335. Westview, 1985.

[47] S. Applequist, G. E. Gahrs, R. L. Pfeffer, and X.-F. Niu, "Comparison of methodologies for probabilistic quantitative precipitation forecasting," *Weather and Forecasting*, vol. 17, no. 4, pp. 783 – 799, 2002.

[48] T. A. Buishand, M. V. Shabalova, and T. Brandsma, "On the choice of the temporal aggregation level for statistical downscaling of precipitation," *Journal of Climate*, vol. 17, no. 9, pp. 1816 – 1827, 2004.

[49] L. D. Monache, T. Nipen, Y. Liu, G. Roux, and R. Stull, "Kalman Filter and Analog Schemes to Postprocess Numerical Weather Predictions," *Monthly Weather Review*, vol. 139, pp. 3554–3570, 2011.

[50] L. D. Monache, F. Anthony Eckel, D. L. Rife, B. Nagarajan, and K. Searight, "Probabilistic Weather Prediction with an Analog Ensemble," *Monthly Weather Review*, vol. 141, pp. 3498–3516, 2013.

[51] S. Rasp and S. Lerch, "Neural Networks for Postprocessing Ensemble Weather Forecasts," *Monthly Weather Review*, vol. 146, pp. 3885–3900, 2018.

[52] A. J. Cannon, "Non-crossing nonlinear regression quantiles by monotone composite quantile regression neural network, with application to rainfall extremes," *Stochastic Environmental Research and Risk Assessment*, vol. 32, pp. 3207–3225, 2018.

[53] J. B. Bremnes, "Ensemble Postprocessing Using Quantile Function Regression Based on Neural Networks and Bernstein Polynomials," *Monthly Weather Review*, vol. 148, pp. 403–414, 2020.

[54] M. Taillardat, O. Mestre, M. Zamo, and P. Naveau, "Calibrated Ensemble Forecasts Using Quantile Regression Forests and Ensemble Model Output Statistics," *Monthly Weather Review*, vol. 144, pp. 2375–2393, 2016.

[55] M. Taillardat and O. Mestre, "From research to applications - examples of operational ensemble post-processing in France using machine learning," *Nonlinear Processes in Geophysics*, vol. 27, no. 2, pp. 329–347, 2020.

[56] W. E. Chapman, A. C. Subramanian, L. Delle Monache, S. P. Xie, and F. M. Ralph, "Improving Atmospheric River Forecasts With Machine Learning," *Geophysical Research Letters*, vol. 46, pp. 10627–10635, 2019.

[57] S. Vannitsem, J. B. Bremnes, J. Demaeyer, G. R. Evans, J. Flowerdew, S. Hemri, S. Lerch, N. Roberts, S. Theis, A. Atencia, Z. B. Bouall?gue,

J. Bhend, M. Dabernig, L. D. Cruz, L. Hieta, O. Mestre, L. Moret, I. O. Plenkovi?, M. Schmeits, M. Taillardat, J. V. den Bergh, B. V. Schaeybroeck, K. Whan, and J. Ylhaisi, "Statistical postprocessing for weather forecasts: Review, challenges, and avenues in a big data world," *Bulletin of the American Meteorological Society*, vol. 102, no. 3, pp. E681 – E699, 2021.

[58] S. Scher and G. Messori, "Predicting weather forecast uncertainty with machine learning," *Quarterly Journal of the Royal Meteorological Society*, vol. 144, no. 717, pp. 2830–2841, 2018.

[59] P. Du, "Ensemble machine learning-based wind forecasting to combine nwp output with data from weather station," *IEEE Transactions on Sustainable Energy*, vol. 10, no. 4, pp. 2133–2141, 2019.

[60] A. Fanfarillo, B. Roozitalab, W. Hu, and G. Cervone, "Probabilistic forecasting using deep generative models," *GeoInformatica*, vol. 25, 01 2021.

[61] H. Demolli, A. S. Dokuz, A. Ecemis, and M. Gokcek, "Wind power forecasting based on daily wind speed data using machine learning algorithms," *Energy Conversion and Management*, vol. 198, no. July, p. 111823, 2019.

[62] K. U. Jaseena and B. C. Kovoor, "A hybrid wind speed forecasting model using stacked autoencoder and LSTM," *Journal of Renewable and Sustainable Energy*, vol. 12, no. 2, 2020.

[63] C. Yu, Y. Li, Y. Bao, H. Tang, and G. Zhai, "A novel framework for wind speed prediction based on recurrent neural networks and support vector machine," *Energy Conversion and Management*, vol. 178, no. July, pp. 137–145, 2018.

[64] R. Li and Y. Jin, "A wind speed interval prediction system based on multi-objective optimization for machine learning method," *Applied Energy*, vol. 228, no. July, pp. 2207–2220, 2018.

[65] L. Wang, X. Li, and Y. Bai, "Short-term wind speed prediction using an extreme learning machine model with error correction," *Energy Conversion and Management*, vol. 162, no. January, pp. 239–250, 2018.

[66] X. Zhao, J. Liu, D. Yu, and J. Chang, "One-day-ahead probabilistic wind speed forecast based on optimized numerical weather prediction data," *Energy Conversion and Management*, vol. 164, no. August 2017, pp. 560–569, 2018.

[67] J. Zhao, J. Wang, Z. Guo, Y. Guo, W. Lin, and Y. Lin, "Multi-step wind speed forecasting based on numerical simulations and an optimized stochastic ensemble method," *Applied Energy*, vol. 255, no. August, p. 113833, 2019.

[68] H. Wang, S. Han, Y. Liu, J. Yan, and L. Li, "Sequence transfer correction algorithm for numerical weather prediction wind speed and its application in a wind power forecasting system," *Applied Energy*, vol. 237, no. December 2018, pp. 1–10, 2019.

[69] T. Dunstan, A. Lock, and A. Skea, "Empirical error correction and feature identification for long term wind resource assessment using support vector regression," *Journal of Renewable and Sustainable Energy*, vol. 8, 2016.

[70] M. Ding, H. Zhou, H. Xie, M. Wu, Y. Nakanishi, and R. Yokoyama, "A gated recurrent unit neural networks based wind speed error correction model for short-term wind power forecasting," *Neurocomputing*, vol. 365, pp. 54–61, 2019.

[71] A. Khoshrou and E. Pauwels, "Short-term scenario-based probabilistic load forecasting: A data-driven approach," *Applied Energy*, vol. 238, pp. 1258–1268, 03 2019.

[72] A. Mashlakov, T. Kuronen, L. Lensu, A. Kaarna, and S. Honkapuro, "Assessing the performance of deep learning models for multivariate probabilistic energy forecasting," *Applied Energy*, vol. 285, 3 2021.

[73] J. Dumas, A. Wehenkel, D. Lanaspeze, B. Cornélusse, and A. Sutera, "A deep generative model for probabilistic energy forecasting in power systems: normalizing flows," *Applied Energy*, vol. 305, 1 2022.

[74] N. Bokde, A. Feijóo, D. Villanueva, and K. Kulat, "A review on hybrid empirical mode decomposition models for wind speed and wind power prediction," *Energies*, vol. 12, 1 2019.

[75] J. Shi, X. Qu, and S. Zeng, "Short-term wind power generation forecasting: Direct versus indirect arima-based approaches," *International Journal of Green Energy*, vol. 8, pp. 100–112, 1 2011.

[76] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems.* O'Reilly Media, Inc., 2nd ed., 2019.

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

[78] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[79] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, pp. 1735–1780, 11 1997.

[80] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *CoRR*, vol. abs/1906.02691, 2019.

[81] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[82] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2014.

[83] B. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964.

[84] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer series in statistics, Springer, 2009.

[85] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[86] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[87] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[88] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, Sept. 2020.

[89] Wes McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference* (Stéfan van der Walt and Jarrod Millman, eds.), pp. 56 – 61, 2010.

[90] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[91] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp,

G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[92] B. Smith, H. Link, G. Randall, and T. McCoy, "Applicability of nacelle anemometer measurements for use in turbine power performance tests: Preprint," 5 2002.

[93] N. Cutler, H. Outhred, and I. Macgill, "Using nacelle-based wind speed observations to improve power curve modeling for wind power forecasting," *Wind Energy*, vol. 15, pp. 245–258, 03 2012.

[94] J. Freedman, J. Zack, J. Schroeder, B. Ancell, K. Brewster, S. Basu, V. Banunarayanan, I. Flores, and E. Ela, "The Wind Forecast Improvement Project (WFIP): A Public/Private Partnership for Improving Short Term Wind Energy Forecasts and Quantifying the Benefits of Utility Operations – the Southern Study Area.," tech. rep., U.S. Department of Energy, 2014.

[95] W. Wang, C. Bruyère, M. Duda, J. Dudhia, D. Gill, M. Kavulich, K. Keene, M. Chen, H.-C. Lin, J. Michalakes, S. Rizvi, X. Zhang, J. Berner, H. Soyoung, and K. Fossell, "ARW Users Guide V3.9-3," 2017.

[96] S.-Y. Hong, Y. Noh, and J. Dudhia, "A New Vertical Diffusion Package with an Explicit Treatment of Entrainment Processes," *Monthly Weather Review*, vol. 134, pp. 2318–2341, 2006.

[97] J. S. Kain, "The Kain–Fritsch Convective Parameterization: An Update," *Journal of Applied Meteorology*, vol. 43, pp. 170–181, 2004.

[98] F. Chen and J. Dudhia, "Coupling an Advanced Land Surface – Hydrology Model with the Penn State – NCAR MM5 Modeling System. Part I : Model Implementation and Sensitivity," *Monthly Weather Review*, pp. 569–585, 2001.

[99] J. Dudhia, "Numerical Study of Convection Observed during the Winter Monsoon Experiment Using a Mesoscale Two-Dimensional Model," *Journal of Atmospheric Sciences*, vol. 46, no. 20, pp. 3077–3107, 1989.

[100] E. J. Mlawer, J. Taubman, P. D. Brown, M. J. Iacono, and S. A. Clough, "Radiative transfer for inhomogeneous atmospheres: RRTM, a validated correlated-k model for the longwave," *Journal of Geophysical Research*, vol. 102, no. D14, 1997.

[101] J. E. Pleim, "A Combined Local and Nonlocal Closure Model for the Atmospheric Boundary Layer. Part I : Model Description and Testing," *Journal of Applied Meteorology and Climatology*, vol. 46, pp. 1383–1395, 2007.

[102] J. W. Tukey *et al.*, *Exploratory data analysis*, vol. 2. Reading, Mass., 1977.

[103] M. Marčiukaitis, I. Žutautaitė, L. Martišauskas, B. Jokšas, G. Gecevičius, and A. Sfetsos, "Non-linear regression model for wind turbine power curve," *Renewable Energy*, vol. 113, pp. 732–741, 2017.

[104] K. E. Taylor, "Summarizing multiple aspects of model performance in a single diagram," *Journal of Geophysical Research: Atmospheres*, vol. 106, pp. 7183–7192, 4 2001.

[105] J. E. Matheson and R. L. Winkler, "Scoring rules for continuous probability distributions," *Management Science*, vol. 22, no. 10, pp. 1087–1096, 1976.

[106] H. Hersbach, "Decomposition of the Continuous Ranked Probability Score for Ensemble Prediction Systems," *Weather and Forecasting*, vol. 15, pp. 559–570, Oct. 2000.

[107] J. L. Anderson, "The impact of dynamical constraints on the selection of initial conditions for ensemble predictions: low-order perfect model results," *Monthly Weather Review*, 1997.

[108] L. D. Monache, J. P. Hacker, Y. Zhou, X. Deng, and R. B. Stull, "Probabilistic aspects of meteorological and ozone regional ensemble forecasts," *Journal of Geophysical Research Atmospheres*, vol. 111, 12 2006.

[109] "Combining spatial statistical and ensemble information in probabilistic weather forecasts," *Monthly Weather Review*, vol. 135, pp. 1386–1402, 4 2007.

[110] T. Gneiting, L. I. Stanberry, E. P. Grimit, L. Held, and N. A. Johnson, "Assessing probabilistic forecasts of multivariate quantities, with an application to ensemble predictions of surface winds," *Test*, vol. 17, pp. 211–235, 2008.

[111] T. Gneiting and A. E. Raftery, "Strictly proper scoring rules, prediction, and estimation," *Journal of the American Statistical Association*, vol. 102, no. 477, pp. 359–378, 2007.

[112] M. Scheuerer and T. M. Hamill, "Variogram-based proper scoring rules for probabilistic forecasts of multivariate quantities*," *Monthly Weather Review*, vol. 143, pp. 1321–1334, 2005.

[113] O. Talagrand, R. Vautard, and B. Strauss, "Evaluation of probabilistic prediction systems," in *Workshop on Predictability, 20-22 October 1997*, (Shinfield Park, Reading), pp. 1–26, ECMWF, ECMWF, 1997.

[114] J. L. Anderson, "A method for producing and evaluating probabilistic forecasts from ensemble model integrations.," *Journal of Climate*, 1996.

[115] T. M. Hamill and S. J. Colucci, "Evaluation of eta–rsm ensemble probabilistic precipitation forecasts," *Monthly Weather Review*, 1998.

[116] D. B. Stephenson and F. J. Dolas-Reyes, "Statistical methods for interpreting monte carlo ensemble forecasts," *Tellus A: Dynamic Meteorology and Oceanography*, vol. 52, no. 3, pp. 300–322, 2000.

[117] I. T. Joliffe and D. B. Stephenson, eds., *Forecast Verification: A Practitioner's Guide in Atmospheric Science, Second Edition*. Wiley, 2012.

[118] C. Wu, J. Wang, X. Chen, P. Du, and W. Yang, "A novel hybrid system based on multi-objective optimization for wind speed forecasting," *Renewable Energy*, vol. 146, pp. 149–165, 2020.

[119] X. Mi, H. Liu, and Y. Li, "Wind speed prediction model using singular spectrum analysis, empirical mode decomposition and convolutional support vector machine," *Energy Conversion and Management*, vol. 180, pp. 196–205, 2019.

[120] H. Liu and Z. Duan, "Corrected multi-resolution ensemble model for wind power forecasting with real-time decomposition and bivariate kernel density estimation," *Energy Conversion and Management*, vol. 203, p. 112265, 2020.

[121] H. Liu, C. Chen, X. Lv, X. Wu, and M. Liu, "Deterministic wind energy forecasting: A review of intelligent predictors and auxiliary methods," *Energy Conversion and Management*, vol. 195, pp. 328–345, 2019.

[122] D. Zhang, X. Peng, K. Pan, and Y. Liu, "A novel wind speed forecasting based on hybrid decomposition and online sequential outlier robust extreme learning machine," *Energy Conversion and Management*, vol. 180, pp. 338–357, 2019.

[123] N. Sun, J. Zhou, L. Chen, B. Jia, M. Tayyab, and T. Peng, "An adaptive dynamic short-term wind speed forecasting model using secondary decomposition and an improved regularized extreme learning machine," *Energy*, vol. 165, pp. 939–957, 2018.

[124] H. Liu, X. Mi, and Y. Li, "Smart deep learning based wind speed prediction model using wavelet packet decomposition, convolutional neural network and convolutional long short term memory network," *Energy Conversion and Management*, vol. 166, pp. 120–131, 2018.

[125] Z. Qian, Y. Pei, H. Zareipour, and N. Chen, "A review and discussion of decomposition-based hybrid models for wind energy forecasting applications," *Applied Energy*, vol. 235, pp. 939–953, 2019.

[126] S. R. Moreno and L. dos Santos Coelho, "Wind speed forecasting approach based on singular spectrum analysis and adaptive neuro fuzzy inference system," *Renewable Energy*, vol. 126, pp. 736–754, 2018.

[127] K. Dragomiretskiy and D. Zosso, "Variational mode decomposition," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 531–544, 2014.

[128] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. 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*, vol. 454, no. 1971, pp. 903–995, 1998.

[129] J. Gilles, "Empirical wavelet transform," *IEEE Transactions on Signal Processing*, vol. 61, pp. 3999–, 08 2013.

[130] I. Daubechies, J. Lu, and H.-T. Wu, "Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 243–261, 2011.

[131] P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol, "Extracting and Composing Robust Features with Denoising Autoencoders," tech. rep., Dept. IRO, Université de Montréal, 2008.

[132] Y. Le Cun and F. Fogelman-Soulié, "Modèles connexionnistes de l'apprentissage," *Intellectica. Revue de l'Association pour la Recherche Cognitive*, pp. 114–143, 1987.

[133] P. Gallinari, Y. Le Cun, S. Thiria, and F. Fogelman-Soulié, "Mémoires associatives distribuées: une comparaison," 1987.

[134] L. Gondara, "Medical image denoising using convolutional denoising autoencoders," in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pp. 241–246, 2016.

[135] E. M. Grais and M. D. Plumbley, "Single channel audio source separation using convolutional denoising autoencoders," in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 1265–1269, 2017.

[136] G. Montavon, W. Samek, and K. R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Processing*, vol. 73, pp. 1–15, 2018.

[137] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Computation*, vol. 4, pp. 1–58, 01 1992.

[138] A. H. Murphy and R. L. Winkler, "A general framework for forecast verification," *Monthly Weather Review*, vol. 115, no. 7, pp. 1330 – 1338, 1987.

[139] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems* (C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, eds.), vol. 28, Curran Associates, Inc., 2015.

[140] H. Fu, C. Li, X. Liu, J. Gao, A. Celikyilmaz, and L. Carin, "Cyclical annealing schedule: A simple approach to mitigating kl vanishing," 2019.