UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA MECÂNICA

JONAS MACHADO MIGUEL

Machine Learning-based Spatio-Temporal Forecasting of Wind Power Generation

São Paulo
2020

JONAS MACHADO MIGUEL

Machine Learning-based Spatio-Temporal Forecasting of Wind Power Generation

Versão original

Relatório preliminar apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do título de Bacharel em Engenharia Mecânica pelo Departamento de Engenharia Mecânica.

Área de concentração: Métodos de Aprendizado de Máquina.

Orientador: Prof. Dr. Fábio Gagliardi Cozman

Coorientador: Dr. Alexandre Cristovão Maiorano

São Paulo

2020

Catalogação-na-publicação

Miguel, Jonas
Machine Learning-based Spatio-Temporal Forecasting of Wind Power Generation / J. Miguel -- São Paulo, 2020.
80 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecânica.

1.Aprendizado de Máquina 2.Energia Eólica 3.Predição 4.Estatística I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecânica II.t.

Abstract

MIGUEL, Jonas Machado. Machine Learning-based Spatio-Temporal Forecasting of Wind Power Generation. 2020. Bachelor of Science Thesis – Escola Politécnica, Universidade de São Paulo, São Paulo, 2020.

Forecasting the behavior of systems in which both temporal and spatial dependencies is of paramount importance across many fields ranging from neuroscience, epidemiology, criminology and transportation. We review the state-of-the-art for spatio-temporal forecasting methods and implement selected approaches for predicting wind power generation at the district-level in Germany. Besides hourly time series for power generation in individual districts in 2000-2015, the analysis considers design and installation specifications for single wind turbines. The models are evaluated on unmodelled periods and locations and benchmarked against conventional statistical time series forecasting methods.

Keywords: Time Series Analysis. Spatio-Temporal Forecasting. Machine Learning, Neural Networks, Wind Power.

Resumo

MIGUEL, Jonas Machado. Machine Learning-based Spatio-Temporal Forecasting of Wind Power Generation. 2020. Bachelor of Science Thesis – Escola Politécnica, Universidade de São Paulo, São Paulo, 2020.

Predizer o comportamento de sistemas regidos por correlações temporais e espaciais é uma tarefa a que se tem atribuída crescente importância em diversas áreas de aplicação, desde neurociência, epidemiologia e criminologia a logística e transporte. Neste trabalho, delineamos o estado da arte para métodos de predição espaço-temporal e implementamos uma seleção desses métodos para a predição de geração de energia eólica no nível distrital na Alemanha. Na análise, levamos em conta tanto séries temporais com resolução horária entre 2000 e 2015, como também especificações de projeto e de instalação de turbinas eólicas individuais. Os modelos são avaliados em períodos não modelados e comparados com métodos estatísticos de previsão.

Palavras-Chave: Análise de Séries Temporais, Predição Espaço-Temporal, Aprendizagem de Máquina, Redes Neurais, Energias Renováveis, Energia Eólica.

Contents

CHAPTER 1

Introduction

Phenomena presenting high socio-economical relevance which are governed by complex dependencies of both spatial and temporal nature are found in diverse domains such as epidemiology, criminology, transportation, climate science and astrophysics [1]. Indeed, the ability to describe a system's behavior is most valuable on instances downstream in the arrow of time: forecasting [2]. Accurate, scalable and feasible rule-based forecasting modeling, however, remains elusive in many cases. Especially as ubiquitous and continuous monitoring data become available, data-driven approaches emerge as a promising alternative.

Conventional data-driven approaches alone, however, have often shown to add limited value in spatio-temporal forecasting [3]. A major reason for this limitation lies on the assumptions they rely upon being typically violated in spatio-temporal settings. Stationarity assumption most of the statistical approaches from time series analysis, while earlier machine learning methods assume data instances are independent and identically distributed (i.i.d.) [1]. Recently, deep learning-based approaches have shown to be able to overcome this essentially by (a) modelling both spatial and temporal dependencies and (b) considering spatial similarities in terms less obvious than geographical proximity alone [4–6].

In the context of renewables, accurately estimating power generation ahead of time poses a major obstacle in progressing towards carbon neutrality in power generation. Heavily conditioned on weather and climate, harvesting energy from renewable sources is characterized by intermittency. Wind power generation, for instance, depends primarily on local wind speeds, which heavily vary in both time and space. Climate changes further aggravates this character, as wind speeds variability are expected to increase [7]. Not accurately knowing how much wind power will be harvested in a certain time and region means power providers have to rely on unnecessarily larger safety margins provided by conventional power plants for ensuring sufficient power supply. This ultimately hampers the

expansion of wind farms and represents therefore a loss for the society, as part of the paid overall generated power is lost, as well as for the environment, as less environment-friendly power sources have to be relied upon [8].

For countries committed to large-scale initiatives such as the *Energiewende* in Germany, this poses a major hindrance in decreasing overall carbon footprint in a sustainable fashion. Accuracy on wind power generation forecasting hence has significant impact on both socio-economical and environmental aspects, in both short and long terms.

1.1 Problem Statement

In spatio-temporal problems, observations of a variable of interest over neighboring locations present not only temporal but also spatial dependencies. While local time series can be predicted individually using conventional univariate statistical techniques, information contained in their spatial and spatio-temporal correlations represent a potential for improving forecasting accuracies. More sophisticated models that allow capturing these dependencies require however supporting evidence on their potential gains that justify the tipicaly longer development times they entail.

1.2 Our Hypothesis

We hypothesize that, in use cases dominated by spatio-temporal dependencies, significant forecasting performance gains can be achieved by spatio-temporal, multi-variate, Machine Learning-based approaches.

1.3 Our Contribution

First, we delineate the state-of-the-art approaches for temporal and spatio-temporal forecasting in different domains, including statistical, machine learning-based approaches. Second, we apply selected approaches for forecasting weekly regional wind power generation in Germany. By benchmarking against more conventional temporal, univariate, statistical approaches, we investigate to what extent more sophisticated modeling approaches add value in terms of accuracy in the use case of onshore wind power generation.

CHAPTER 2

Background

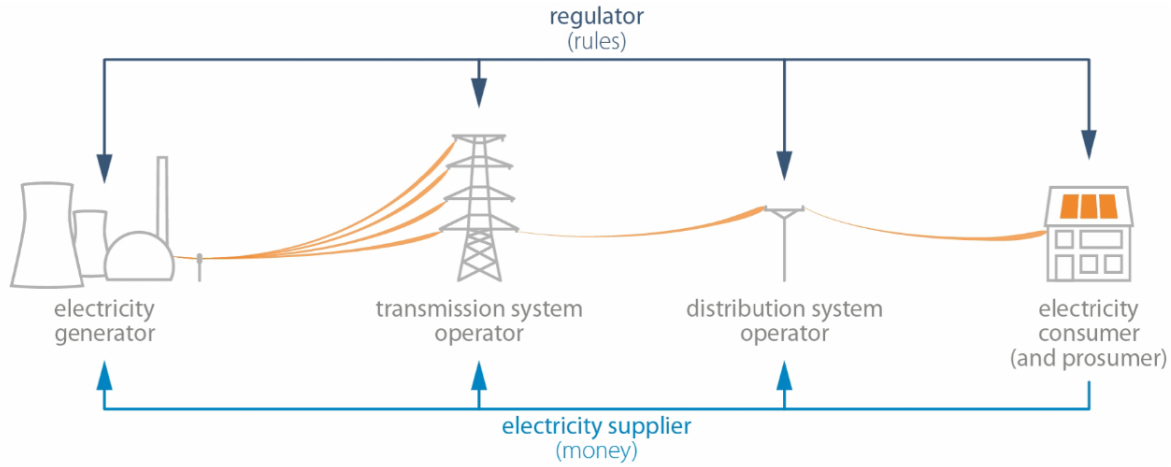
Chapter Overview

- The general structure of the EU liberalized electricity markets: players and roles.
- Harvesting wind power: first principles, main process stages and variables underlying power conversion, the central challenge of intermittency, forecasting as an approach for handling intermittency, major forecasting approaches.
- Time series forecasting: task definition, relevance, tasks categories, typical model requirements, methods and metrics for assessing model generalization performance, key requirements besides accuracy, examples of data-driven forecasting approaches, selecting models from a same approach.
- Spatio-temporal forecasting: enhancing models by going beyond exclusively temporal correlations, examples of approaches, how metrics are changed.

2.1 Liberalized Electricity Markets

In a liberalized electricity market, multiple entities are involved in supplying energy to final consumers, as 2.1 illustrates. In the EU, these parties are electricity generators, transmission system operators (TSO), distribution system operator (DSO), electricity supplier, and regulator [9]. TSOs are responsible for long-distance transport of energy and for balancing supply and demand in timeframes under quarter-hour. Imbalances of this nature cause deviations from the nominal frequency and shortages in more severe cases. DSOs are responsible for delivering electricity to consumers. Electricity suppliers buy energy from generator parties and resell it to consumers. The liberalized EU grid system counts yet with another kind of entity, the balance responsible parties (BRPs), which although not directly involved in neither production nor in consumption activities, are financially responsible for the supply-demand balance [9].

Figure 2.1 – The different stages of electricity supply and the responsible parties in a liberalized market. Adapted from [9].



2.2 Wind Power Generation

In 1920, Betz [10] modeled a generic wind harvesting system as an open-disc actuator and, by using the energy conservation equation for a stream tube flowing through this disk, he derived an upper limit for the power harvested by a horizontal-axis wind turbine. The *Betz Limit*, as it is known, is a function of rotor diameter D (via the rotor swept area A) and the average free stream wind velocity v at hub height H (2.1).

$$P_{ideal} = \frac{1}{2}\rho \cdot A(D) \cdot v^3 \tag{2.1}$$

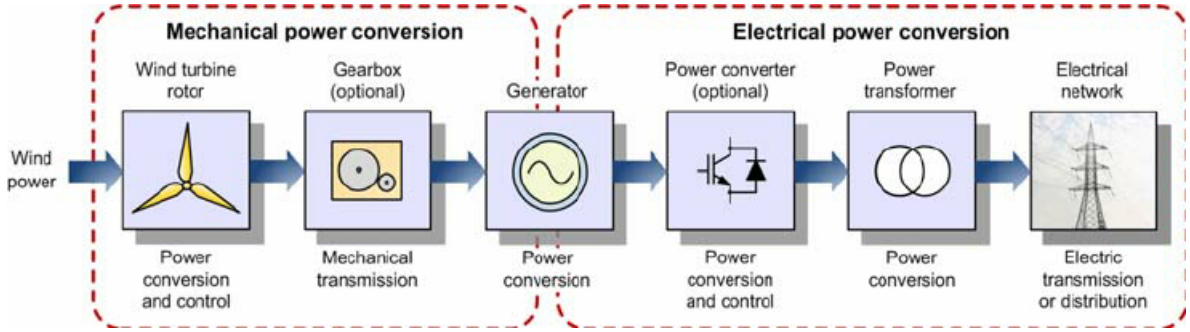
Due to losses such as those associated to (1) momentum deficit in lower atmosphere boundary layer, (2) wakes from neighboring turbines, (3) suboptimal yaw angle and

(4) blade tip vortices, the power harvested by the turbine rotor is only a fraction C_p (coefficient of power) of this idealized maximum. Further losses (a) of mechanical nature in the interfaces rotor-gearbox and gearbox-generator (2.2), (b) of electrical nature in the interface generator-converter are modeled by the fractions η_m and η_e, respectively, to yield the actual power generation as measured at the power converter, 2.2 [11].

$$P = C_p \eta_m \eta_e \cdot \frac{1}{2} \rho \cdot A(D) \cdot v^3 \tag{2.2}$$

In this equation, D and H are design variables. The air density ρ may vary during operation due to changes in air temperature, but its effects are often negligible. Finally, C_p, v depend both on design (e.g., hub height H, blade profiles) and operation conditions (e.g., velocity speed and direction).

Figure 2.2 – The different stages of the overall wind power conversion process. Adapted from [12].



In operation, the dominant source of variability for the generated power is v. Being climate and weather-dependent, it is also the main reason for the intermittency and non-dispatchability of wind power [13]. This dependence motivates the usage by designers and generation operators of the so-called *wind-to-power curves* (or *power curves*), which are semi-empirical relations that allow one to determine the generated power P by knowing the wind velocity v.

As design, planning, operation, maintenance, and trading of wind power are subject to such high variabilities, forecasting wind power generation (WPG) provides value for the different players in the electricity grid, illustrated in 2.1. Table **??** gives some examples of how various system operation aspects can profit from forecasts at different time scales.

Table 2.1 – Forecasting horizons in wind power generation and main applications.

Forecasting Horizon	Definition	Applications
Very Short	$\sim secs\ - 0.5h$	turbine control, load tracking
Short	$0.5h - 72h$	pre-load sharing
Medium	$72h$ –1 $week$	power system management, energy trading
Long	1 $week$ –1 $year$	turbines maintenance scheduling

Power generation from single turbines can also be aggregated at different levels. Market operators, for example, profit the most from regional aggregations, since for energy trading, this resolution is sufficiently high, with higher resolutions across the same space scales of interests often too costly [14]. In countries such as Germany, where continental and national renewables-promoting public funding initiatives such as the *Energiewende* resulted in high penetration of wind power in the grid, forecasting wind power generation accurately has a tangible impact both environmentally and economically.

The intermittency of renewables motivated an alternative measure of power generation: the *capacity factor* (CF). CF is defined as the ratio of the actual generated power and the installed capacity. When considering WPG data across long timespans for both analysis and forecasting, it is usual that new commissionings take place, which manifests itself as a step perturbation into the overall generated power. In this case, CF can be useful as it is mostly insensitive to single new commissionings. Climate and weather-conditioned local wind velocities imply for the power generation not only significant temporal dependencies but also significant spatial dependencies. As air masses influence one another in different scales, wind power generation in neighboring turbines tends to present higher correlations than turbines distant from one another [15]. Therefore, wind power generation is a phenomenon with dominant spatio-temporal dependencies. Usual approaches to forecasting wind power generation are physical, statistical, and machine learning-based [14]. The physical approach relies on the modeling of the power curve using Computational Fluid Dynamic (CFD) models, taking Numerical Weather Prediction (NWP) as inputs for defining the boundary conditions. The main limitations of this approach are (a) the high costs involved in the development of such models, along with (b) the large uncertainties entailed by the NWP data. The statistical approach uses historical data and statistical time series models to produce forecasts for wind speed, which is then used in the power curve for forecasting the power generation itself. Finally, in machine learning approaches, one uses historical data for wind speed or power generation, eventually combined with historical data of weather conditions to forecast either (a) local wind speeds, with their subsequent transformation into generated power via power-curve or (b) generated power directly.

2.3 Time Series Forecasting

In [16], Brockwell & Davis define time series as "a set of observations y_t, each one being recorded at a specific time t." When observations are recorded at discrete times, they are called a discrete-time time series, on which we focus this work.

An important task in time series analysis is time series forecasting, which concerns "the prediction of data at future times using observations collected in the past" [17]. Time series forecasting permeates most aspects of modern business, such as business planning from production to distribution, finance and marketing, inventory control, and customer management [18]. In some business use cases, a single point in forecasting accuracy may represent millions of dollars [19, 20].

Time series forecasting tasks can be categorized in terms of (a) inputs, (b) modeling, and (c) outputs. In terms of inputs, one can use exogenous features or not, one or more input time series (univariate *versus* multivariate). In terms of modeling, one must define a resolution (e.g., hourly, weekly), can aggregate data in different levels (hierarchical *versus* non-hierarchical), and can use different schemes for generating models (we distinguish statistical from machine learning-based). Finally, regarding outputs, a forecasting task might involve making predictions in terms of single values or whole distributions (deterministic *versus* probabilistic), point-predictions or prediction intervals, predict values for either a single point or for multiple points in future time (one-step-ahead *versus* multi-step-ahead). In this work, we focus on deterministic, one-step ahead point forecasting, where one is interested in obtaining a function $f : \mathbb{R}^T \rightarrow \mathbb{R}$ (a *forecasting model*) that maps historical observations $\boldsymbol{y}_{1:T} = \{y_1, \ldots, y_T\}$ of a variable y_t to its value in a future time step $T + h$, for a forecasting horizon of interest h.

The main requirement for a forecasting model concerns the accuracy of its forecasts $\hat{y}_{t|T}$. This accuracy is quantified by a *metric*, which summarizes the distribution of the forecast error $e_t = y_t - \hat{y}_{t|T}$ over the different evaluation timesteps t. In the following subsections, we introduce some typical options for (a) schemes for defining the evaluation timesteps t (2.3.1), (b) accuracy metrics (2.3.2), as well as (c) approaches for generating forecasting models (2.3.3.1, 2.3.3.2, 2.3.3.3).

2.3.1 Model Evaluation

Assessing the performance of a model f requires defining the time indexes t for evaluating the forecast errors e_t . In a naive approach, one could use all available data for both model inference and evaluation. This would, however, result in a highly biased estimate of the model generalization performance. Less biased estimations could be attained instead by partitioning the available dataset into a *training dataset* $\boldsymbol{y}_{1:T} = \{y_1, \ldots, y_T\}$, exclusive for model inference, and a *test dataset* $\boldsymbol{y}_{T+1:T'} = \{y_{T+1}, \ldots, y_{T'}\}$, used for model evaluation

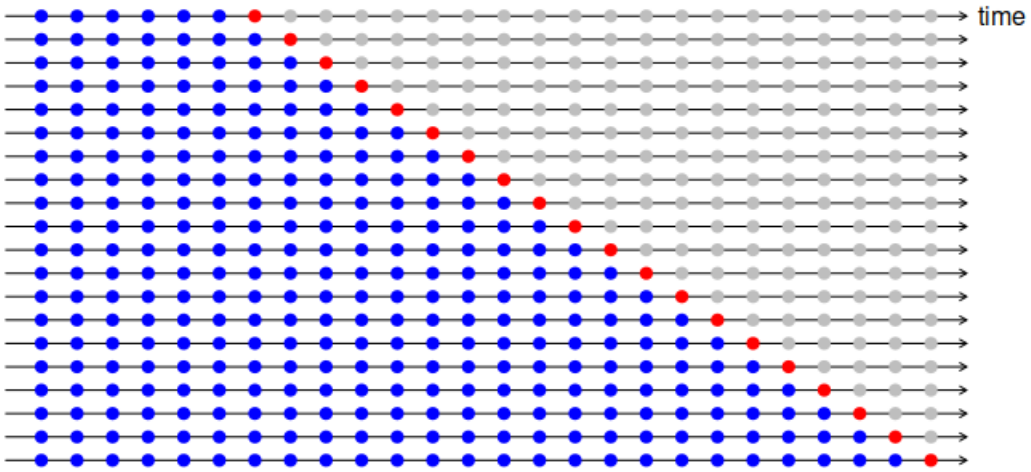
(2.3). Once an estimate for the model performance is attained, a separate model inference using both partitions can be carried out, so that the epistemic part of the generalization error, resulting from limited data in model inference, is kept at a minimum.

Figure 2.3 – Partitioning the available data in training and test datasets (adapted from [21]).

Furthermore, it is necessary that this partitioning results in two sets of successive observations, in order to preserve the *Markovian dependence* underlying the sequential observations. Even under this constraint, however, the choice on what point to split the data is still arbitrary, implying that assessing model performance on a single arbitrary choice would result in a biased estimate. To minimize this bias, the model performance can be assessed for several different splitting points. The partial results are then aggregated, typically by averaging, into an overall result of model performance. This procedure is known as *out-of-sample cross-validation*.

As the forecast error generally increases for longer forecasting horizons, the out-of-sample estimate might overestimate the generalization error, especially if only one-step forecasts are of interest. For overcoming this, only the first point in the test data is used in evaluating the error. This approach is known as *expanding window cross-validation*, and is illustrated in 2.4.

Figure 2.4 – The expanding window cross-validation scheme (adapted from [21]).



2.3.2 Accuracy Metrics

Many different metrics exist, each one summarizing the error distribution in a different way. Some of the most usual definitions are presented from 2.3 to 2.10 (see e.g., [5, 6, 22].

In particular, $MASE$ and $MdRAE$ use as denominator the forecast errors of the naïve model, which takes the last known value to forecast the next point. The naïve model can be shown to be optimal for a random walk process [22].

$$RMSE = \sqrt{\mathbb{E}(e_t^2)} = \sqrt{\frac{1}{(T' - T - 1)} \sum_{t=T+1}^{T'} e_t^2} \tag{2.3}$$

$$MAE = \mathbb{E}(|e_t|) = \frac{1}{(T' - T - 1)} \sum_{t=T+1}^{T'} |e_t| \tag{2.4}$$

$$MAPE = \mathbb{E}(|e_t / y_t| \cdot 100\%) = \frac{100\%}{(T' - T - 1)} \sum_{t=T+1}^{T'} \left| \frac{e_t}{y_t} \right| \tag{2.5}$$

$$sMAPE = \frac{100\%}{T' - T - 1} \sum_{T=T+1}^{T'} \frac{|e_t|}{(|y_t| + |\hat{y}_t|)/2} \tag{2.6}$$

$$MdAPE = q_{0.5}(|e_t / y_t| \cdot 100\%) \tag{2.7}$$

$$sMdAPE = q_{0.5} \left(200\% \cdot \frac{|e_t|}{y_t + \hat{y}_t} \right) \tag{2.8}$$

$$MASE = \mathbb{E} \left(\left| \frac{e_t}{e_{t,naïve}} \right| \right) \tag{2.9}$$

$$MdRAE = q_{0.5} \left(\left| \frac{e_t}{e_{t,naïve}} \right| \right) \tag{2.10}$$

By summarizing the forecast error distribution into a reduced set of values, forecasting metrics are essential in model development as well as in method development. To forecasters (model developers) and forecast users, metrics offer a concise, unambiguous way to communicate accuracy requirements and specifications. For methods developers, it allows comparing different methods across different use cases, forecasting settings, and datasets.

On the one hand, single metrics concisely convey information about the error distribution, which is useful for comparing models and making decisions. On the other hand, a single metric cannot convey all aspects of the error distribution, and often using more than one metric becomes necessary to ensure sufficiency [2]. Therefore, deciding on a group of metrics often involves a trade-off between conciseness and sufficiency.

Metrics differ in interpretability, scale invariance, sensitivity to outliers, symmetric penalization of negative and positive errors, and behavior predictability as $y_t \to 0$ [22]. Therefore, it is important that the choice on the metrics set is coherent with the application requirements [2]. For example, while failing to forecast single sudden peaks in local wind

speed (wind gusts) might not be important in wind farm planning, it might be a primary requirement for wind turbine operation. Table **??** summarizes sensitivities for the presented metrics.

Table 2.2 – Forecasting accuracy metrics and their sensitivities to scale and outliers.

Alias	Name	Scale Sensitivity	Outliers Sensitivity
RMSE	Root Mean Squared Error	●	●
MAE	Mean Absolute Error	●	●
MASE	Mean Absolute Scaled Error	○	●
MAPE	Mean Absolute Percentual Error	○	●
MdAPE	Median Absolute Percentual Error	○	○
sMAPE	Symmetric Mean Absolute Percentual Error	○	○
sMdAPE	Symmetric Median Absolute Percentual Error	○	○
MdRAE	Median Relative Absolute Error	○	○

Although often the most important one, accuracy is often just one of many requirements in a forecasting model development. In [2], Armstrong reports that value inference time, cost savings resulting from improved decisions, interpretability, usability, ease of implementation, and development costs (human and computational resources) tend to be of comparable importance to researchers, practitioners, and decision-makers.

2.3.3 Forecasting Approaches

In general, forecasting approaches, statistical or machine learning-based alike, attain models by minimizing the forecast errors on the training set. This optimization process, often iterative, uses an optimization algorithm to update the model parameters configuration towards one that either (a) maximizes their likelihood or (b) minimizes a loss function on the training set.

The likelihood is, in essence, the relative number of ways that a configuration of model parameters can reproduce the provided data [23]. In contrast, loss functions summarize the distribution of forecast errors, much like accuracy metrics. Loss functions are subject to an additional requirement, however, which is their suitability as objective function in the convex optimization underlying most of model inferencing schemes. Therefore, although it is important that the objective function guiding the model inference is coherent with the metrics used for evaluating the models, they do not have to coincide. The Mean Squared Error (MSE, 2.11) is a typical choice for a loss function for continuous-type responses, as it accounts for both bias and variance errors, besides exhibiting smoothness amenable to

convex optimization [24].

$$MSE = \frac{1}{T}\sum_{t=1}^{T} e_t^2 \tag{2.11}$$

Table 2.3 provides an overview of the approaches reviewed in this work. We start by presenting simple forecasting approaches*, which are often used as baselines for other approaches [17].

Table 2.3 – Forecasting approaches presented in this work. Most of these methods only model dependencies of temporal nature and are presented in this section. Exception are DCRNN, ST-GCN, and Graph WaveNet (ML-based), presented in section 2.4. They explicitly approach a more general forecasting setting where capturing both temporal and spatial dependencies is a central concern.

Statistical		**Hybrid**	**ML-based**	
Exponential Smoothing	SES Holt's LinearHolt-Winter's	ES-RNN	**Temporal**	RNN LSTM N-BEATS
ARIMA	AR MA ARIMA		**Spatio-Temporal**	DCRNN ST-GCN GWNet

2.3.3.1 Baseline Approaches

Naïve method forecasts the signal as constant at its last observed value (2.12). As the naïve forecast is the optimal prediction for a random walk process, it is also known as the *random walk* method.

$$\hat{y}_{T+h|T} = y_T \tag{2.12}$$

Seasonal Naïve method models time series as harmonic with period k observations (i.e., perfectly seasonal with seasonal period k), and for a given point in future, suggest the corresponding last observed value from the last season (2.13). For example, all monthly forecasts for any future June assume the value from the last observed June value.

$$\hat{y}_{T+h|T} = y_{T+h-k} \tag{2.13}$$

Drift method. The forecast for any point assumes a constant value rate of change, with

*Analogous to Murphy in [25], we draw distinctions between the concepts of method, model, and model inference algorithm. A method can specify (1) how training data is used to generate a model (training, model inference, i.e., inference of its parameters) and (2) how a generated model uses its parameters and its input to make a prediction (inference). We denote by a model any unique configuration of parameters in a space defined by a method. Equivalently, a model represents a response surface (deterministic model) or the distribution of the response conditional on its inputs (probabilistic model).

values themselves starting from the latest observed value:

$$\hat{y}_{T+h|T} = y_T + h\left(\frac{y_T - y_1}{T-1}\right). \tag{2.14}$$

Historical Average (HA) method. The forecast for any point assumes a constant value: the average of the historical data (2.12).

$$\hat{y}_{T+h|T} = \frac{1}{T}\sum_{t=1}^{T} y_t \tag{2.15}$$

2.3.3.2 Statistical Approaches

Statistical forecasting approaches are characterized by the modeling of the time series as a realization of a stationary stochastic process [26], [27]. The two most widely used families of statistical methods are the Exponential Smoothing (ES) family and the ARIMA family [17].

In the ES approach, the time series is modeled as combination of interpretable components [26]. In the *classical decomposition* [28], these components are trend component m, seasonal component d, and random noise (*white noise*) ε_t, which are linearly combined to reconstruct the time series:

$$y_t = m_t + s_t + a_t. \tag{2.16}$$

We now describe some of the most known methods from the ES family. **SES (Simple Exponential Smoothing method)** predicts for the next period the forecast value for the previous period, adjusting it using the forecast error (2.17). Parameter: $\alpha \in \mathbb{R}_{[0,1]}$.

$$\hat{y}_{t+1} = \hat{y}_t + \alpha(\hat{y}_t - \hat{y}_{t-1}) \tag{2.17}$$

Holt's Linear method features an additive trend component [29]. Parameters: $(\alpha, \beta^*) \in \mathbb{R}^2_{[0,1]}$

$$\begin{aligned} \hat{y}_{t+h|t} &= \ell_t + b_t h, \\ where \quad \ell_t &= \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \qquad (level) \\ b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \quad (growth) \end{aligned} \tag{2.18}$$

Holt-Winters' method features additive trend and multiplicative seasonality components, for a seasonality length m, and forecasting horizon h. Parameters: $(\alpha, \beta^*, \gamma) \in \mathbb{R}^3_{[0,1]}$

(usual bounds, refer to [29] for details).

$$\begin{aligned}
&\hat{y}_{t+h|t} = (\ell_t + b_t h) s_{t-m+h_m^+}, \\
&where\ \ell_t = \alpha \frac{y_t}{s_{t-m}} + (1-\alpha)(\ell_{t-1} + b_{t-1}) && (level) \\
&\qquad b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*) b_{t-1} && (growth) \\
&\qquad s_t = \gamma y_t / (\ell_{t-1} + b_{t-1}) + (1-\gamma) s_{t-m} && (seasonal)
\end{aligned} \tag{2.19}$$

ARIMA (Autoregressive Integrated Moving Average) methods [30] rely on repeatedly applying a difference operator to the observed values until the differenced series resemble a realization of some stationary stochastic process [26]. We denote by $\nabla^k(\cdot)$ the difference operator of order k. For $k = 1$, $\nabla y_t = y_t - y_{t-1}$; for $k = 2$, we have $\nabla^2(y_t) = \nabla(\nabla y_t) = \nabla y_t - \nabla y_{t-1} = y_t - 2y_{t-1} + y_{t-2}$ and so forth. Another operator useful in ARIMA methods is the *backshift operator* $B^k(\cdot)$ with lag k. For $k = 1$, we have $By_t = y_{t-1}$. For $k = 2$, $B^2(y_t) = B(B(y_t)) = y_{t-2}$.

AR (Autoregressive) method. Linear regression with past values of the same variable (lagged values) as predictors. A constant level c and a white noise $\varepsilon_t \sim WN(\mu_\varepsilon, \sigma_\varepsilon^2)$ are considered. Parameters: $\boldsymbol{\phi} = [\phi_1\ \phi_2\ \cdots\ \phi_p]^\top,\ \mu_\varepsilon,\ \sigma_\varepsilon, c$. Hyperparameter: p.

$$\hat{y}_t = c + \varepsilon_t + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} \tag{2.20}$$

MA (Moving Average) method. Linear regression with lagged forecast errors $\varepsilon_\tau = \hat{y}_\tau - y_\tau$ as predictors. Parameters:$\boldsymbol{\theta} = [\theta_1\ \theta_2\ \cdots\ \theta_q]^\top,\ \mu_\varepsilon,\ \sigma_\varepsilon, c$. Hyperparameter: q.

$$\hat{y}_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + ... + \theta_q \varepsilon_{t-q} \tag{2.21}$$

(Non-seasonal) ARIMA method. Linear regression, with lagged *differenced* values y'_τ and lagged errors as predictors. It combines autoregression on the differenced time series with a moving average model, hence the name *Autoregressive Integrated Moving Average*, with *integration* referring to the reverse operation of differencing, used when reconstructing the original time series from its differenced version. Parameters: $\boldsymbol{\phi} = [\phi_1\ \phi_2\ \cdots\ \phi_p]^\top, \boldsymbol{\theta} = [\theta_1\ \theta_2\ \cdots\ \theta_q]^\top,\ \mu_\varepsilon,\ \sigma_\varepsilon, c$. Hyperparameters: p, d, q.

$$\hat{y}'_t = c + \varepsilon_t + \phi_1 y'_{t-1} + ... + \phi_p y'_{t-p} + \cdots + \theta_1 \varepsilon_{t-1} + ... + \theta_q \varepsilon_{t-q} \tag{2.22}$$

2.3.3.3 Machine Learning Approaches

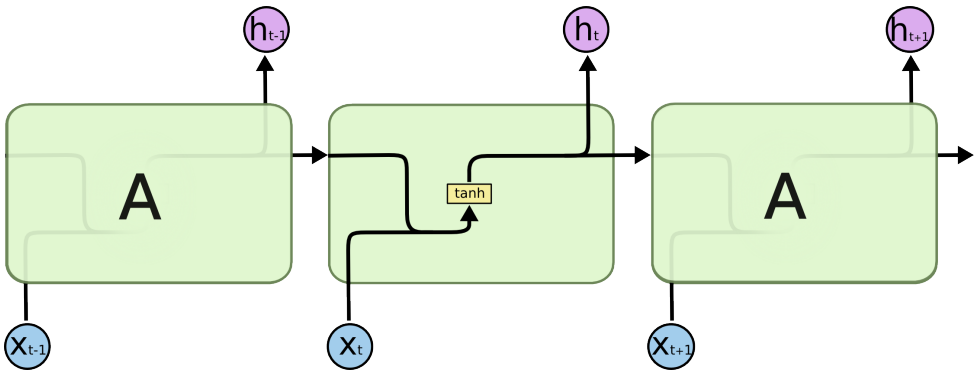
Approaches solely based on Machine Learning struggled until recently to consistently outperform statistical time series forecasting approaches [31]. Despite relying on biased evidence (e.g., models were evaluated across all time series without any sound choice nor

search for hyperparameters), Makridakis claimed in [31] that "hybrid approaches and combinations of methods are the way forward for improving the forecasting accuracy and making forecasting more valuable." Oreshkin et al. challenged in [18] this conclusion, introducing N-BEATS, a pure deep learning method that was shown to outperform statistical and hybrid methods, while also ensuring interpretability of intermediate outputs.

Below we present selected deep learning methods helpful for understanding current state-of-the-art approaches for both wind power generation-specific applications and in general univariate time series forecasting applications.

RNN (Recurrent Neural Network) uses the recurrent layer as building block: a cell that updates its state according to (a) its previous state h_{t-1} and (b) its current input x_t (2.5). By performing this update at every timestep of a time series, this basic structure allows the RNN to express temporal dependencies in time series. An RNN can be built by serializing several of these self-looping cells between the input layer and the output layer for achieving higher-order mappings and thus capturing more complex temporal dependencies. The major limitation of RNN in its basic design (recurrent layer as in 2.5) is its inability to capture dependencies that exist across longer periods than a few timesteps. It arises from a phenomenon called *vanishing gradients*: while inferring optimal parameters via gradient descent (learning phase), the gradients calculated via backpropagation through time become too small to guide the optimization.

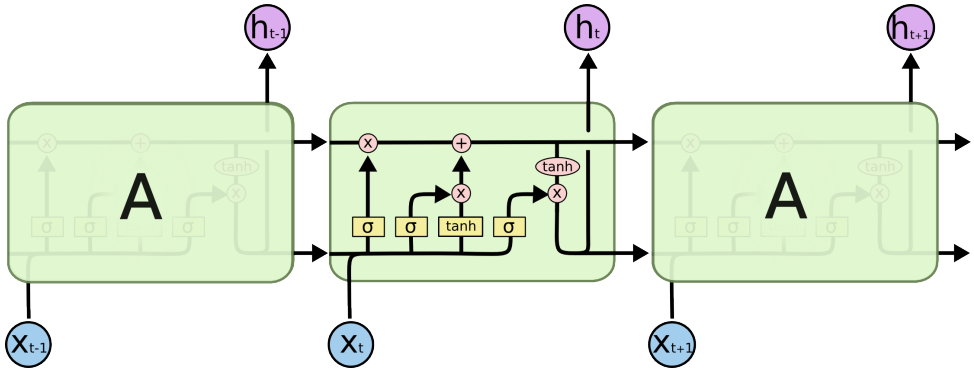
Figure 2.5 – The basic RNN architecture in its unfolded representation. Arrows indicate transfers of input and hidden states (adapted from [32]. Every block concatenates the last hidden state with the current input, passing the result to an activation function (tanh in this illustration). The result is carried forward as the updated hidden state.



LSTM (Long-Short Term Memory) is a type of RNN, and improves on its basic design most importantly by including an long memory state which is allowed to be transferred across several update steps with only minimal changes (superior horizontal line inside the repeating module in 2.6. This allows information to persist across many cell updates, thus making it possible to capture long-term dependencies. The extent to which this long memory state is preserved is controlled by forget gate, illustrated in 2.6) by the leftmost

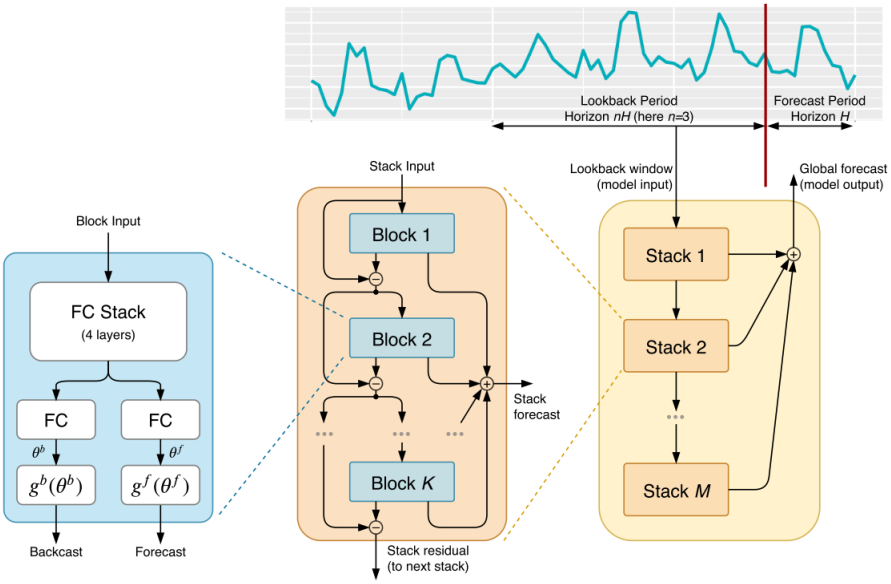
vertical path inside the cell. The other paths represent other gated state transfers, which determine how (a) the previous cell state, (b) the previous long memory state and (c) the current cell inputs are combined and passed to the next cell iteration and as input to deeper layers.

Figure 2.6 – Basic LSTM architecture in its unfolded representation (adapted from [32]).



NBEATS uses as building block (a) a multi-layer fully connected network with ReLU nonlinearities, which feed (b) basis layers that generate a backcast and a forecast output. Blocks are arranged into stacks, organized to form a model (2.7. Models resulting from this architecture consistently outperformed state-of-the-art methods for univariate forecasting across different horizons and thousands of time series datasets of different nature, while using a single hyperparameter configuration [18].

Figure 2.7 – NBEATS architecture (adapted from [18]).



2.3.3.4 Hybrid Approaches

Hybrid methods combine machine learning and statistical approaches by using the outputs from statistical engines as features [18]. Below we present ES-RNN, a hybrid method winner of the 2017 M4 forecasting competition.

ES-RNN. It uses Holt-Winters' ES method as statistical engine for capturing the seasonal and level components from the time series into features, which are then used by an LSTM model to exploit non-linear dependencies.

$$\begin{aligned} \hat{y}_{t+h|t} &= LSTM(y_t, \ell_t, s_t) \\ where \quad \ell_t &= \alpha \frac{y_t}{s_{t-m}} + (1-\alpha)\ell_{t-1} \qquad (level) \\ s_t &= \gamma y_t / \ell_t + (1-\gamma) s_{t-m} \qquad (seasonal) \end{aligned} \tag{2.23}$$

2.3.4 Model Selection

As models have parameters, so do methods have their own, often referred to as *hyper-parameters*. They may control the space of model parameters configurations, the model inference process or eventually the loss function [33]). Hyperparameters may have a major influence on the performance of resulting models. Accounting for this effect requires yet another partition in order to attain a minimally unbiased estimate of the resulting generalization error. When working with three partitions, one for model inference, another for assessing its generalization error given a hyperparameters configuration and another one for assessing it across different hyperparameters configurations, authors often refer to them as training, validation and test set, respectively.

2.4 Spatio-Temporal Forecasting

In the spatio-temporal (ST) version of this problem, one aims to attain a function $f : \mathbb{R}^{|V| \times T} \to \mathbb{R}^{|V|}$ that maps historical observations of a quantity across different regions which can be interpreted as graph vertices $v \in V$, $\boldsymbol{y}_t = [y_{1,t} \;\; y_{2,t} \;\; \cdots \;\; y_{|V|,t}]^\top$, to its value $\boldsymbol{y}_{t+1}$ in the next timestep (2.24).

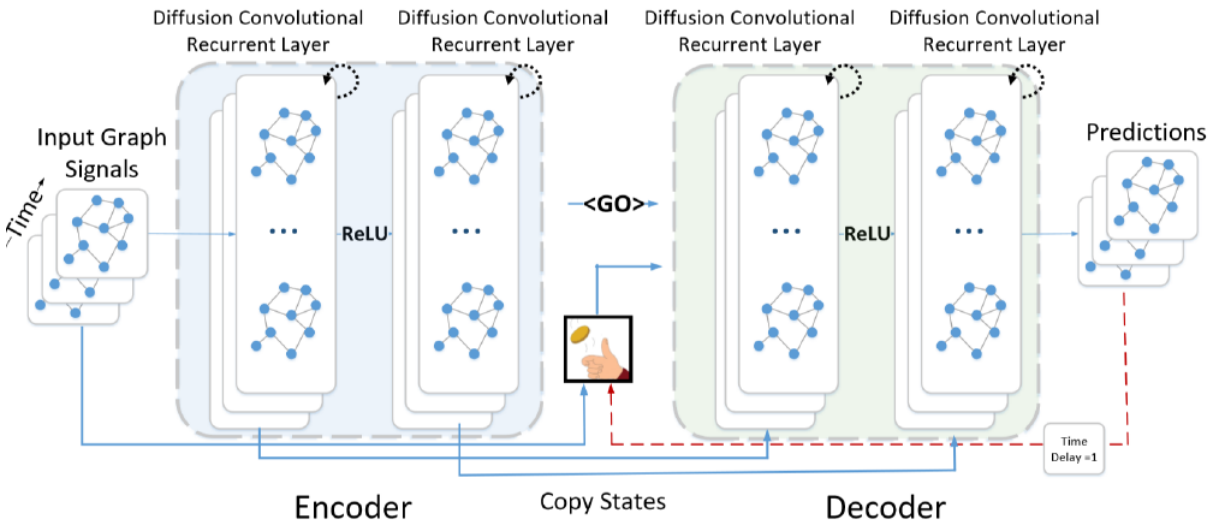
$$[\boldsymbol{y}_{t-T+1}, \; \cdots \; , \boldsymbol{y}_t] \xrightarrow{f(\cdot)} \boldsymbol{y}_{t+1} \tag{2.24}$$

For some forecasting problems such as for the weather-conditioned wind power generation, the spatial dependency might play an important along with the temporal dependencies themselves [15]). In this work, we consider three different approaches to the ST forecasting problem. In a naïve approach, time series for different locations are modeled independently,

thus neglecting spatial dependencies. In a second approach, the time series are modeled jointly via a multivariate forecasting approach, where for generating a single model one relies on historical observations not from a single but from several input variables, which can be expressed by a sequence of input vectors $\boldsymbol{X}_{1:T} = \{\boldsymbol{X}_1, ..., \boldsymbol{X}_T\}$. Finally, we consider the explicit modeling of both spatial and temporal dependencies via dynamic graphs. The latter approach is represented by the methods presented below.

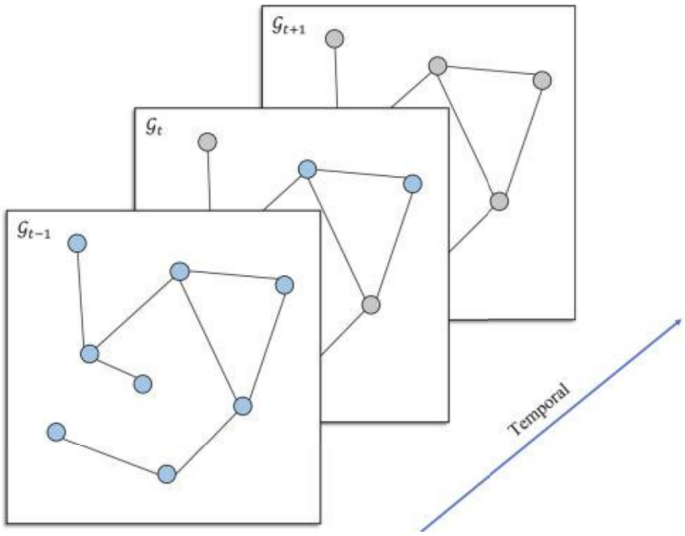
DCRNN (Diffusion Convolutional RNN). RNN is leveraged by replacing the matrix multiplication by a diffusion convolution [34]). Motivated by the traffic forecasting problem, where spatial dependencies are directional (non-Euclidean), Li et al. [4]) recast the spatio-temporal evolution of a variable as a diffusion process on a directed graph, where every node corresponds to a sensor. Learning is performed via (1) diffusion convolution, further integrated with a (2) seq-to-seq learning framework, and a (3) scheduled sampling for modeling long-term dependencies (2.8).

Figure 2.8 – The DCRNN architecture (adapted from [4]).



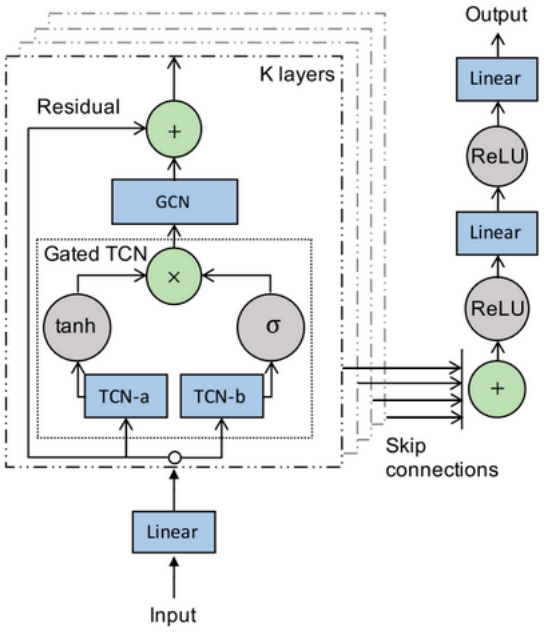
ST-GCN. A spatial-temporal graph is generated by stacking graph frames from all timesteps, each frame representing the graph state at a specific time (2.9). The spatial-temporal graph is partitioned, and to each of its nodes is assigned a weight vector. Finally, a graph convolution is performed on the weighted spatial-temporal graph.

Figure 2.9 – ST-GCN underlying principle (adapted from [5]).



GWNet. The Graph Wavenet uses as building blocks a Temporal Convolution Network (TCN) and a Graph Convolution Network (GCN) for capturing spatio-temporal dependencies in every module. A core idea is the usage of a learnable self-adaptative adjacency matrix, which allows node dependencies to change over time and not necessarily be determined by their distances [6, 34].

Figure 2.10 – The Graph Wavenet architecture (adapted from [6]).



2.4.1 Accuracy Metrics

The usual accuracy metrics in spatio-temporal forecasting are similar to their counterparts in the temporal setting, the main difference concerning the aggregation in space over the

$v \in |V|$ regions. We present some of them in 2.4, 2.5, 2.3.

$$MAE = \frac{1}{|V|(T' - T - 1)} \Sigma_{v=1}^{|V|} \Sigma_{t=T+1}^{T'} |\hat{\boldsymbol{y}}_t^{(v)} - \boldsymbol{y}_t^{(v)}| \tag{2.25}$$

$$MAPE = \frac{100\%}{|V|(T' - T - 1)} \Sigma_{v=1}^{|V|} \Sigma_{t=T+1}^{T'} \frac{|\hat{\boldsymbol{y}}_t^{(v)} - \boldsymbol{y}_t^{(v)}|}{|\boldsymbol{y}_t^{(v)}|} \tag{2.26}$$

$$RMSE = \sqrt{\frac{1}{|V|(T' - T - 1)} \Sigma_{v=1}^{|V|} \Sigma_{t=T+1}^{T'} (\hat{\boldsymbol{y}}_t^{(v)} - \boldsymbol{y}_t^{(v)})^2} \tag{2.27}$$

CHAPTER 3

Use Case

Chapter Overview

- Requirements.
- Data resources available.
- Exploratory Data Analysis: investigating the generation process, major patterns in the datasets and their limitations.
- Implications of the use case peculiarities to the data pipeline design.

3.1 Requirements

The most valuable kind of information they require for driving decisions of balance responsible parties (BRPs) are wind forecasts for daily power generation in subregional scale. Therefore, we focus this work on these resolution requirements: daily, districtwise forecasts. With regards to scale, we focus on week-ahead, country-wide predictions. We set the localization of interest for these forecasts as weeks in June 2015 in Germany.

This time period is known to be less susceptible to wind gusts and other weather anomalies in the region. By doing this, we avoid models being evaluated on untypical or abnormal settings, which we prioritize for comparing overall *expectations* of performances of models arising from different forecasting approaches. We choose this country due to the availability and quality of its wind power generation datasets [35]. Furthermore, we limit our scope to point forecasting predictions, although uncertainty quantification could potentially add more value and even be a key requirement for these players in practice.

3.2 Datasets

For all investigations in this work, we use the two datasets from [35]. We name them the measurements dataset and the sensors dataset. The first dataset is the main source of temporal information, while the second conveys most of the spatial information we use.

The measurements dataset (fig. 3.1) consists of measurements for wind power generation (in kW) in Germany, being measured, aggregated and reported by wind farm operators and published by german state governments.

Table 3.1 – Measurements dataset

t	DE145	DE114	...	DE12A
2000-01-01 00:00	19.3	538.1	...	176.7
2000-01-01 01:00	37.6	513.6	...	6.8
...	...	...	...	...
2015-12-31 23:59	0.8	2.9	...	1.8

The sensors dataset (fig. 3.2) reports individual turbines design and commissioning specifications provided by operators, most importantly the geolocation, the rated power and the commissioning date.

Both datasets were compiled and prepared by authors in [35], most importantly by imputing missing entries (about 15% in the measurements dataset, 8% in the sensors dataset) via machine learning-based methods.

Table 3.2 – Sensors dataset

id	power	dt	hubheight	diameter	NUTS_ID	lat	lon
1	1500	2002-06-01	61.5	77.0	DE145	9.628	48.532
2	1500	2002-06-01	61.5	77.0	DE145	9.636	48.533
...	...	...	...	...	...	...	...
26119	3050	2015-09-25	120.0	116.8	DEG0M	12.345	50.824

3.3 Exploratory Data Analysis

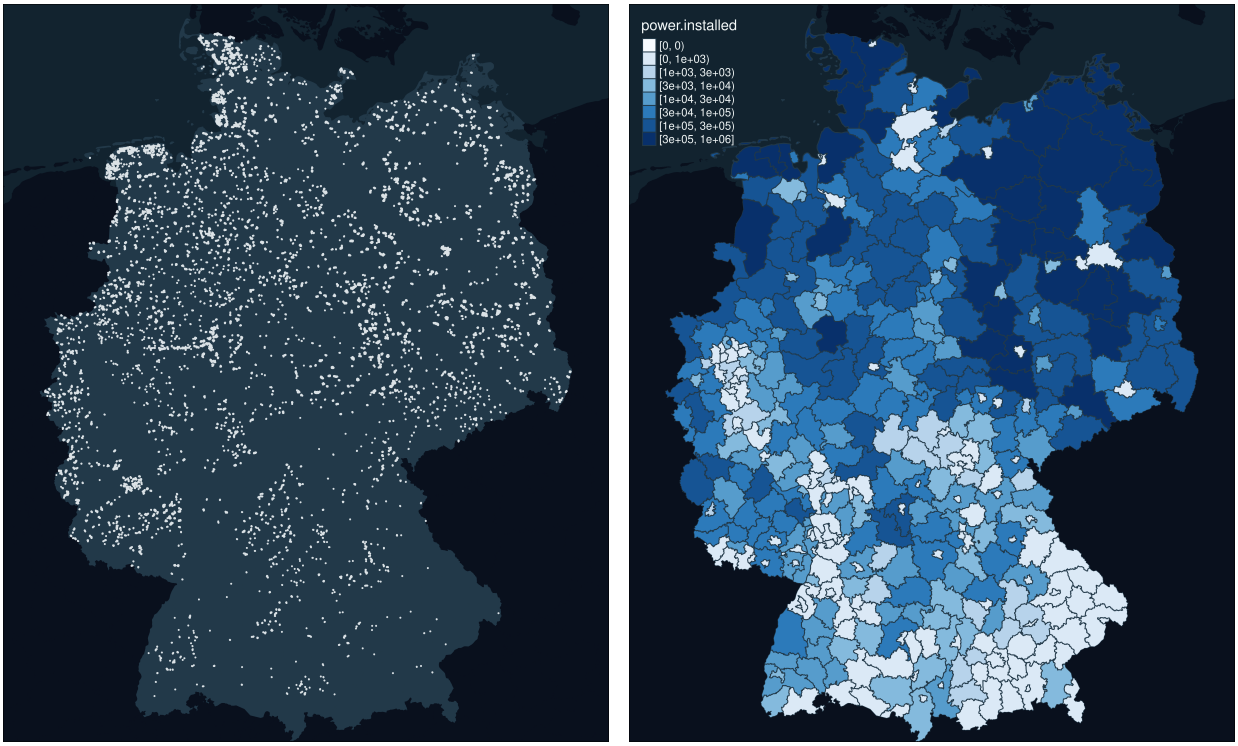
We carried out an exploratory data analysis on both datasets to better understand their generation process, as well as to identify data patterns and limitations. As final purpose, we used the findings resulting from this analysis to not only guide our design decisions on both preprocessing and modeling, but also to define our key modeling assumptions.

First, we identified major spatial and temporal conditions underlying the data generation. Figure 3.1 highlights the most relevant findings. Figure 3.1a shows how the 23191 onshore wind turbines are spatially distributed across 296 districts harvesting wind power 3.1b. The northern region still concentrates most of the generation capacity (fig. 3.1b) and concentrates most of the largest units so as to harness the local higher wind power availability [36]. Evidence for a contrasting recent trend to this can be seen in figure 3.1c. Over the last decade, Germany has been commissioning more midsized turbines closer to locations of high electricity demand at the southern and central Germany so as to prevent curtailments due to network congestion [15].

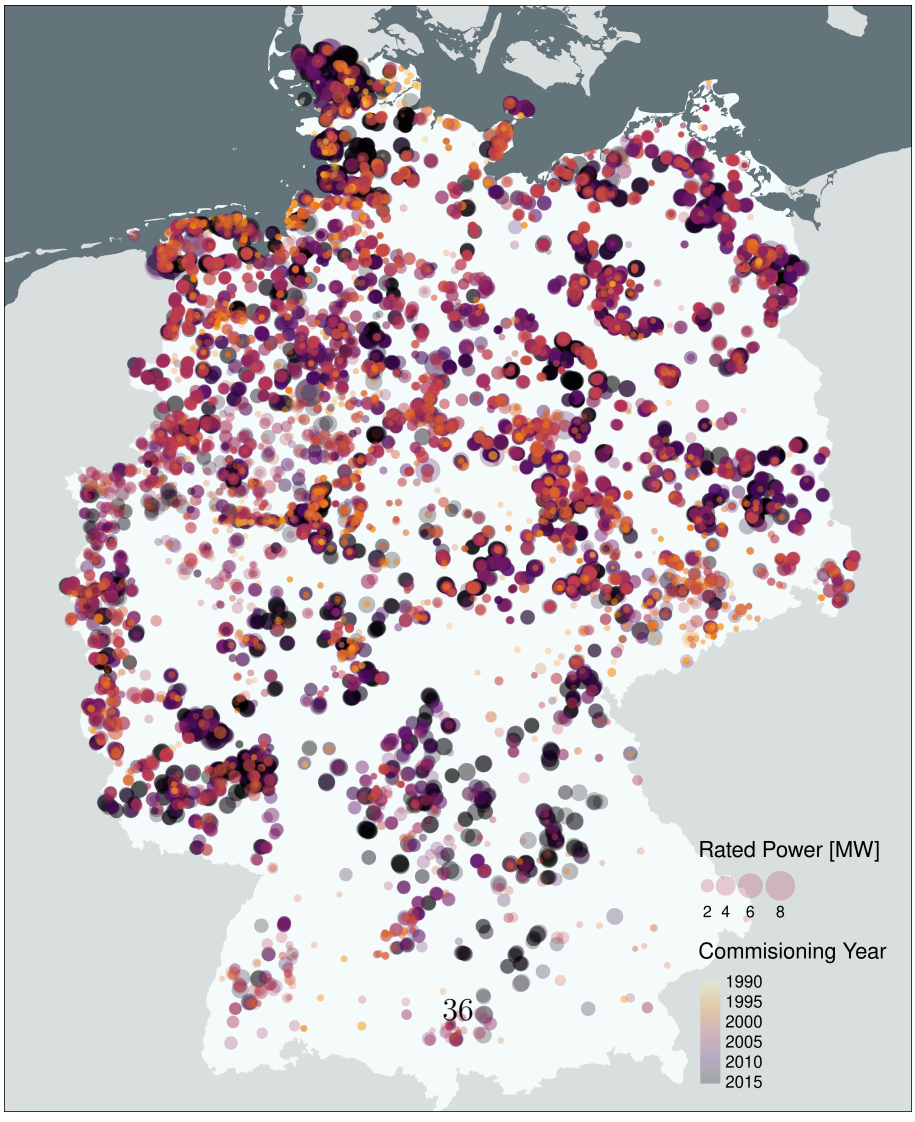
Figure 3.1 – Spatio-temporal distribution of wind power generation capabilities.

(a) Turbines geodistribution.

(b) Installed capacity by district.

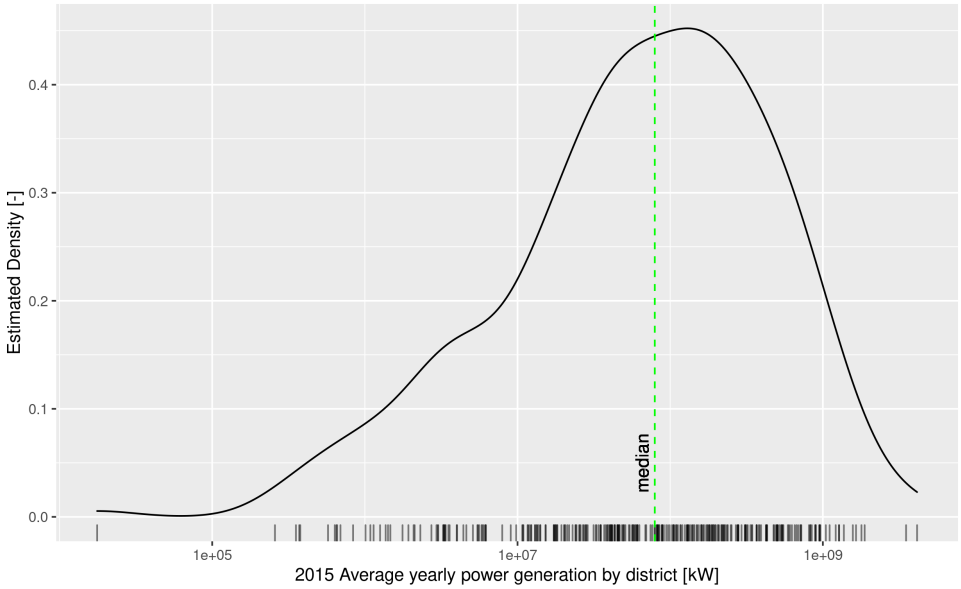


(c) Geodistribution over time of rated power in new commissionings.



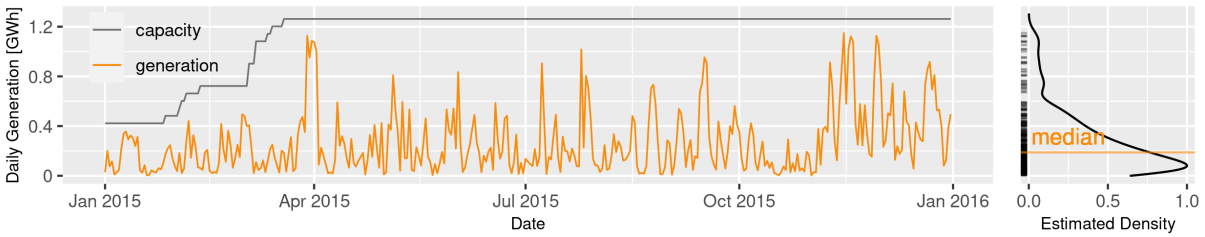
The second step in our analysis concerned the determination of (1) how power generation is distributed and (2) what is a typical production behavior. The kernel-estimated density function for the districtwise, yearly average power generation (e.g. figure 3.2) suggests a unimodal, approximately log-normal distribution.

Figure 3.2 – Distribution of yearly districtwise wind power generation.



We noticed near-median yearly power productions tend to be distributed across time as the one shown in figure 3.3. The observed Weibull distribution is in agreement with accounts in the literature [15, 37, 38]. Here, we notice an important behavior trend: not only the measurements values vary, but also their amplitude, as new turbines are commissioned (and decommissioned). This is specially significant in the case of Germany, where wind power generation rapidly increases its share in the energy portfolio.

Figure 3.3 – A typical power generation (in kW) in a year, as represented by the district of Bernkastel-Wittlich (DEB22).



This might pose a significant limitation to models inferred from historic data for power generation alone, as they would be unable to capture the correlations arising from the causal effect of (de-)commissionings on future values of power generation. In other

words, trained on the dataset as it is, models would be unable to account for eventual sudden increases in power generation due to new commissionings, eventually incurring in drastically underestimating forecasts. Furthermore, we would expect this effect to be more pronounced for longer forecast horizons, as the probability of new commissionings for the forecast period would increase.

One way to address this would be to provide to the models an exogenous feature which informed it about new commissioning ahead. Not every forecasting approach supports this, however, as in the case of historical average or single-input ARIMA variants. This would thus limit the comparability of methods performance.

For this reason, we use another approach in this work. Essentially, we train the models to predict a normalized version of the time series and handle the effect by re-scaling model outputs in a model-agnostic, post-processing step. More specifically, we scale each time series value by the installed capacity for the specific district and point in time. In the renewables field, the resulting scaled variable is known as the Capacity Factor (CF). The resulting time series are illustrated in figures 3.4 and 3.5.

Figure 3.4 – A typical power generation (in Capacity Factor) in a year, as represented by the district of Bernkastel-Wittlich (DEB22).

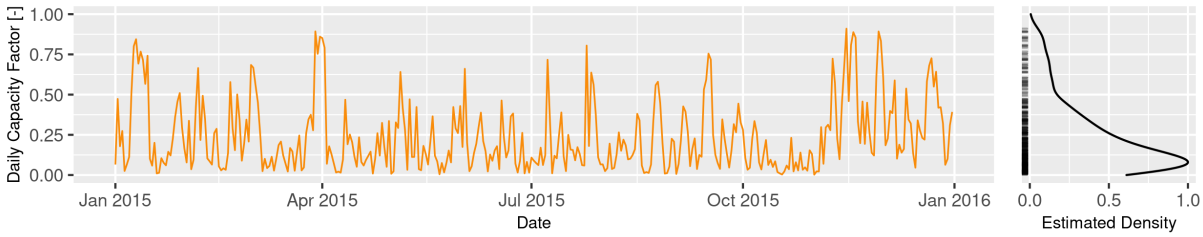
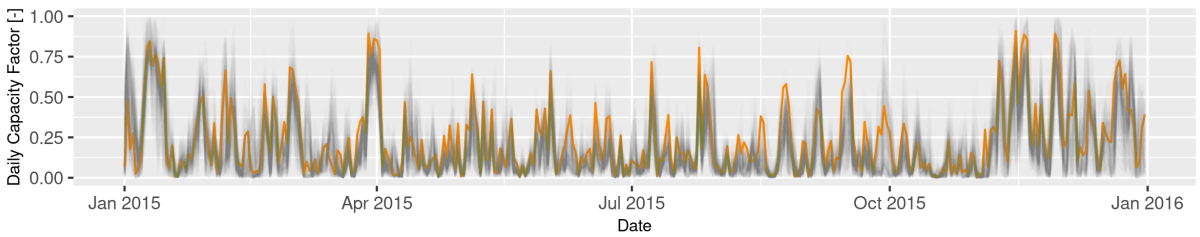


Figure 3.5 – Power generation in terms of CF for all districts. The highlighted CF curve refers to Bernkastel-Wittlich (DEB22).

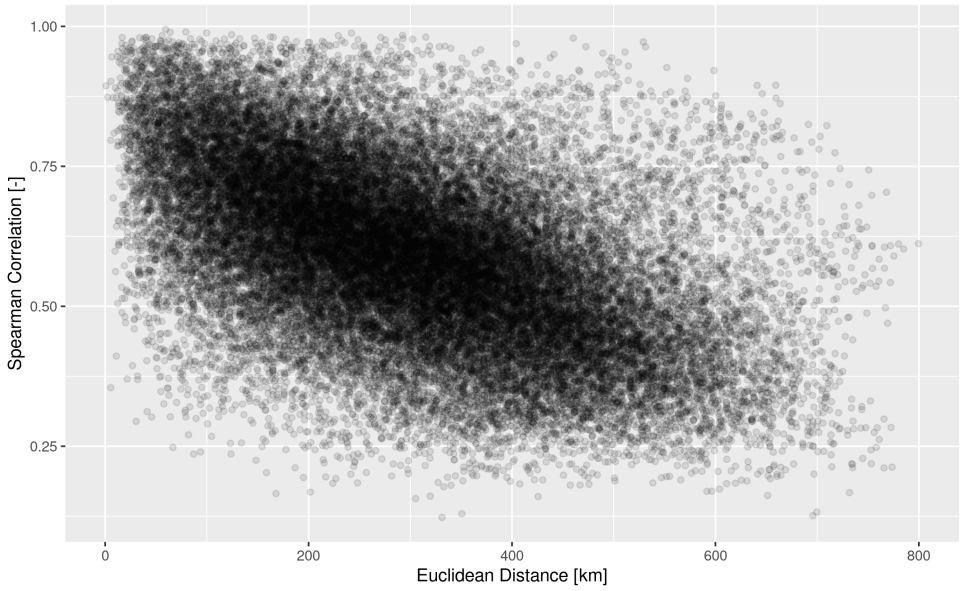


While the transformation of time series from kW into Capacity Factors solves the issue concerning new commissionings, another issue involving the distribution of time series values remains. Being Weibull-distributed, values in a time series would be concentrated around the median, thus less discernible from one another than if they were normally

distributed, for instance. Thus, we expect gains in terms of informational entropy and thus in model training cost and performance by properly scaling the model inputs. The nature of the Weibull density function suggests that any linear scale such as the min-max scaling would not suffice to improve discernibility (informational entropy) in our data.

In a third step of our exploratory data analysis, we investigated to what extent power generation is correlated in space and time. For the spatial dependency, we inquired *"how more similarly do closer districts behave than distant ones?"*. We performed this by assessing how pairwise Spearman correlations between districts change as districts are more distant from one another [15]. We verified that pairwise Spearman correlations between closer districts are significantly higher than distant ones, evidencing a significant spatial character for power production (3.6). In fact, districts present significant correlations ($\rho_s > 0.8$) for distances up to about 150 km. This *decorrelation distance* is in agreement with values found for Central Europe by other authors [15].

Figure 3.6 – Spearman Correlations for all $nCR(296, 2)$ pairs of districts power generation (in CF) time series.



We followed a similar procedure to assess the temporal correlation between time series. In order to attain evidence for this, we used cross-correlograms, which evaluate a cross-correlation function such as the Pearson coefficient ρ as one time series is shifted in time in relation to the other time series. Figure 3.7 shows an instance of correlogram for a pair of districts at decorrelation distance. Figure 3.8 presents the superposition of all correlograms. We verify a decorrelation time delay of 12 hours, which is in agreement with the literature for Central Europe [15].

Figure 3.7 – Cross-correlogram for the districts sample pair (DEB22, DEE145).

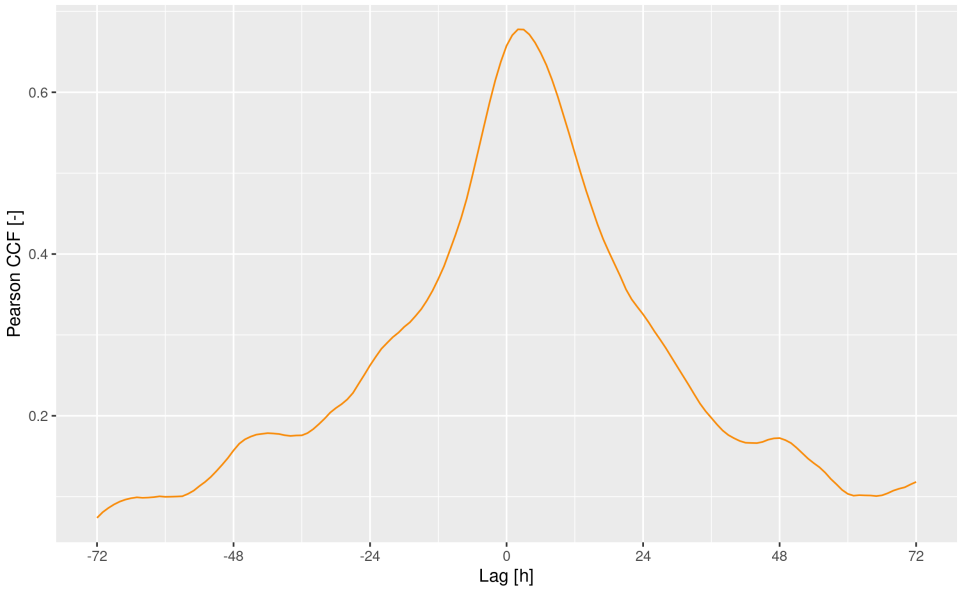
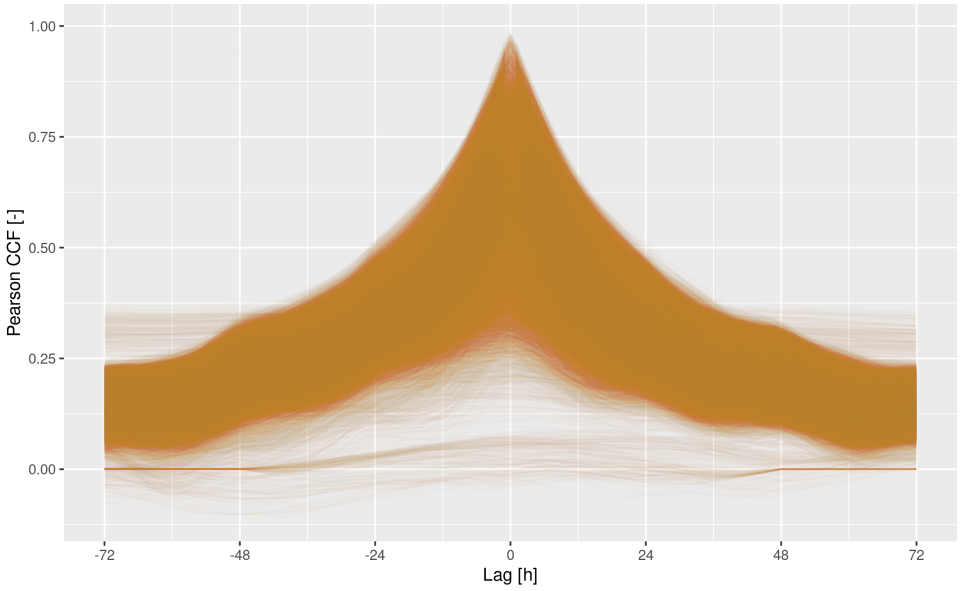


Figure 3.8 – Superposition of cross-correlograms for all $nCR(296, 2)$ pairs of districts.



3.4 Consequences for the data pipeline design

In general, the EDA was important to verify that spatio-temporal dependencies are significant in the use case at hand. Below, we summarize the main consequences this analysis had to our design decisions.

Preprocessing. Handling missing data is not necessary, but scaling the time series e.g. into capacity factors is expected to heavily influence model performances. Weibull-

distributed time series might require non-linear scaling.

Modeling. Aggregating time series to time resolutions coarser than 12 hours might diminish correlations between them, thus potentially limiting the potential accuracy gains in using spatio-temporal approaches.

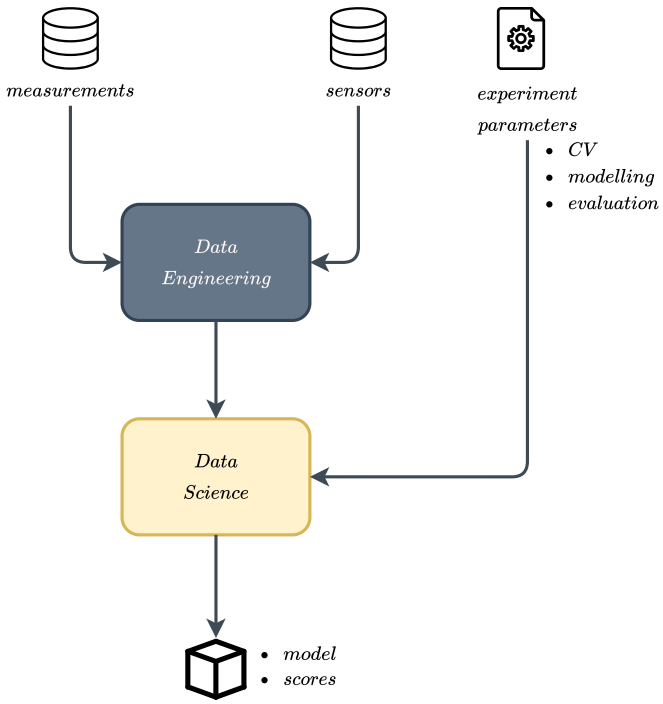
Key modeling assumptions. In this work, we assume negligible the effects of (a) curtailments, (b) maintenance of individual units, (c) decommissioning of individual units, (d) sudden increases in capacity factor due to technological advances.

CHAPTER 4

Data Pipeline

We developed a pipeline (fig. 4.1) comprising (1) a data engineering pipeline, and (2) a data science pipeline). We devoted significant part of our effort into following existing and developing best practices in data science development standards, so as to ensure reproducibility of results as well as reusability of methods and tools. One of the key Python frameworks we used to attain that is the McKinsey's Kedro [39].

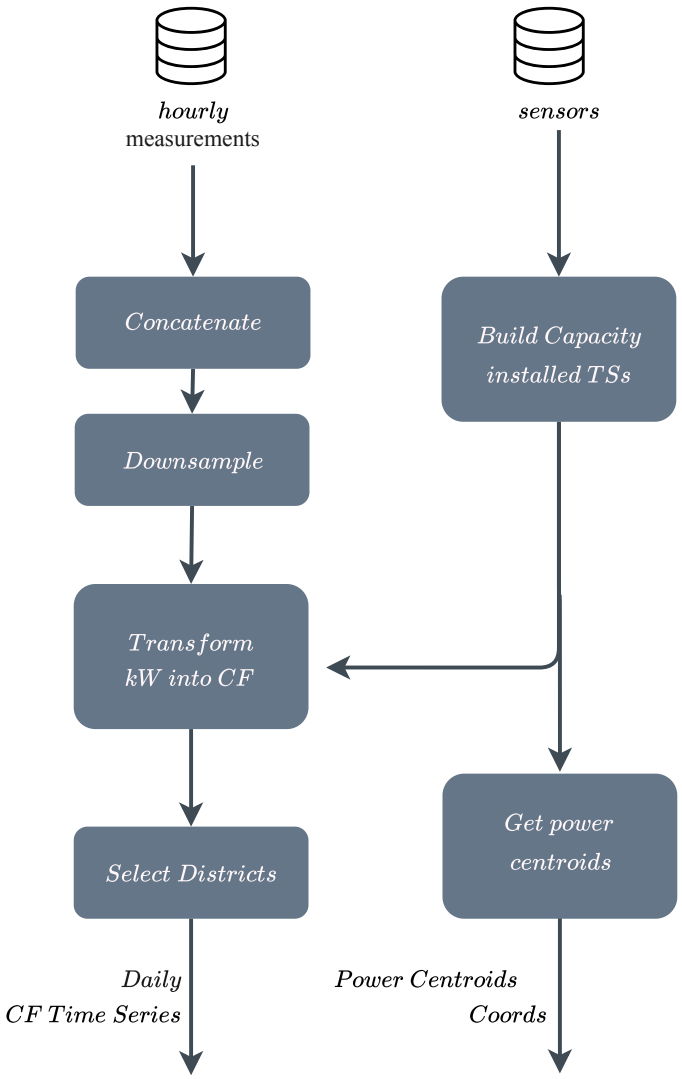
Figure 4.1 – The project pipeline.



4.1 Data Engineering Pipeline

The data engineering pipeline (fig. 4.2) encompasses all the data processing steps involved between (a) measurements and sensors datasets and (b) daily capacity factors, sensors graph inputs.

Figure 4.2 – The data engineering pipeline.



Get Capacity Installed Time Series. We load the sensors dataset and build a time series for the capacity installed in every district, essentially by grouping turbine entries by district, and performing a cumulated sum of power ratings over the commissioning date-sorted entries.

Get Power Centroids. A centroid position is defined for every district, not by its bari-center, but from the rated power-weighted average of all its single turbines coordinates. In

practice, the power centroid changes its position every time a new turbine is commissioned. We neglect this variation over time, and take the resulting average as sufficiently accurate for its purpose. Namely, we use the power centroids for calculating representative Euclidean distances between districts, which we used for the EDA and also when calculating the adjacency matrix initialization values for the graph-based spatio-temporal forecasting methods.

Concatenate and Downsample. We concatenate the measurements dataset (power generated districtwise, in kW), which is provided for every year, into a single hourly dataframe, then downsample it into a daily measurements by summing same-day entries.

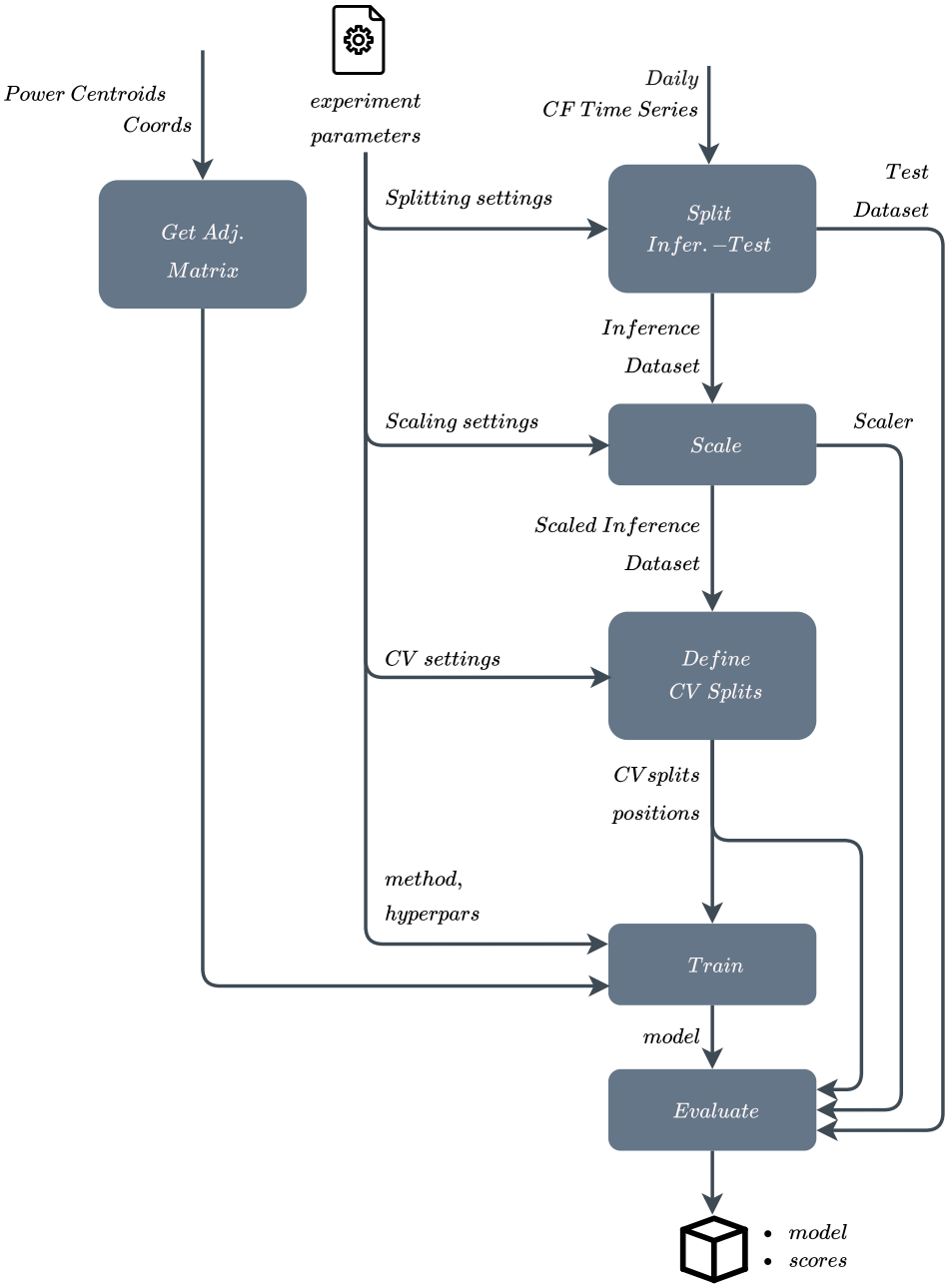
Filter Districts. We filter out previously determined districts which either represent outliers in spatial correlogram (3 from 303) or have zero installed capacity by 2015-01-01 (4 from 303). Also, districts in which by 2000-01-01 a single turbine represents more than 50% of its installed capacity (18 districts in total). Although the latter measure might represent a deviation from the industry use case, we perform it in favor of metrics representativity, as otherwise models overall performance metrics would be biased by low predictibility of ill-conditioned time series e.g. from districts where wind harvesting are still in early phases.

Transform kW to CF. Every entry in measurements time series dataset is normalized by the corresponding local installed capacity at the same day. This results in daily time series for capacity factors in every district.

4.2 Data Science Pipeline

The data science pipeline (fig. 4.3) processes (a) user-defined parameters, (b) the capacity factors dataset resulting from the data engineering pipeline and, when required, also (c) the power centroids.

Figure 4.3 – The data engineering pipeline.



Get Initial Adjacency Matrix. We calculate the matrix of the pairwise Euclidean distances and transform it into an adjacency matrix. For initializing the values of the

(constant) adjacency matrix in DCRNN or the self-adaptive adjacency matrix in the Graph WaveNet case, we follow the procedure in [4, 6] apply the Gaussian Kernel on the distances matrix, so that $A_{ij} = exp(-D_{ij}^2/2\sigma_D^2)$, where σ_D is the standard deviation of the distances matrix. Also as in [**?**], we promote sparsity in the adjacency matrix for computational efficiency by thresholding the entries in the adjacency matrix. However, instead of defining an arbitrary threshold value for A_{ij}, we prune adjacency values when the distances they are calculated from surpasses the decorrelation distance of 150 km. This node function is only processed in experiments with DCRNN and Graph WaveNet, as they are the only methods considered which rely on an adjacency matrix.

Split Inference-Test Data. We split the capacity factors measurements dataset into a model inference dataset, used for training and model selection, and a test dataset, reserved for the model performance evaluation. The split is done according to the user-defined date ranges defined for each partition.

Scale. We apply the user-defined sequence of scaling and offsetting methods on the inference data.

Define CV Splits Positions. Date ranges for defining respectively the training and the validation datasets are determined, the validation window always positioned on dates later than training window last entry. User-defined entries for this function include the cross-validation scheme and the pertaining parameters. For expanding window CV, the parameters are the relative size of the shortest train window (0.0 - 1.0 proportion of model inference dataset size), the number of total CV passes, the number of steps ahead, and the forecast window size.

CV Train. Trains a model for every CV split as well as one for the entire scaled model inference dataset. In the case of single time series-modeling methods such as Holt-Winters Exponential Smoothing, the resulting model is actually a simple collection of single time series submodels.

Evaluate. Makes predictions using every trained model in the experiment and calculate overall model performance metrics.

CHAPTER 5

Experiments Settings

Currently, all forecasting approaches are evaluated on a reduced case consisting of (a) 5 districts on northern Germany (DEF0C, DEF07, DEF0B, DEF05, DEF0E) located within 80 km distance from one another, (b) model inference time window from 2013-01-01 to 2015-06-22 and test time window from 2015-06-23 to 2015-06-29. We chose the test time window to be in a year period known to be less susceptible to wind gusts and other weather anomalies. Both models were evaluated in terms of predictions in capacity factors, with cross-districts uniform average of metrics. With regards of model tuning, only manual procedure has been carried out.

HW-ES. For preprocessing, relies on quantile transformation into a normal distribution, followed by an offsetting by the absolute value of the minimum of every scaled time series. The latter step is performed to ensure model inputs are strictly positive so as to allow for multiplicative seasonal approach in the HW-ES method. As for hyperparameters, we use additive trend, multiplicative seasonal, seasonal period of 7 steps (days).

GWNet. For preprocessing, relies on a Z-standard scaling. As for hyperparameters, we define most importantly the number of nodes (5), the sequence length (12), the learning rate (1E-3), and the learning decay rate (0.97).

CHAPTER 6

Results

Table 6.1 summarizes the models performances, according to cross-district uniform averages of metrics. For the reduced case, the spatio-temporal approach GWNet generally outperforms the purely temporal approach HW-ES. In a larger-scale study case including more districts, we expect the gap between the models to increase, as GWNet can make use of more inter-time series correlations. The current very limited reduced case does not allow us to draw definite conclusions on approaches capabilities beyond this, so further comparisons on larger-scales cases are necessary.

Table 6.1 – Performance comparison of different approaches for wind power generation in reduced case.

Metric	**HW-ES**	**GWNet**
MAE	0.182	**0.091**
RMSE	**0.116**	0.131
MAPE	45.4%	**1.78**%

CHAPTER 7

Conclusion and Next Steps

Preliminary results for the reduced case suggest that accounting for cross-time series correlations might indeed improve model accuracy.

Besides expanding the use case to the other districts and its training time window, we are reassessing the plausibility of the model evaluation period (currently a fixed week) and metrics. We intend to reassess model performance in terms of predictions in power generation in kW, and to make a separate model evaluation with metrics more appropriate for the specific use case of renewables power generation.

Bibliography

[1] Gowtham Atluri, Anuj Karpatne, and Vipin Kumar. Spatio-Temporal Data Mining: A Survey of Problems and Methods. *ACM Computing Surveys*, 51(4), 11 2017.

[2] J Scott Armstrong. PRINCIPLES OF FORECASTING: A Handbook for Researchers and Practitioners. Technical report, 2002.

[3] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLoS ONE*, 13(3), 3 2018.

[4] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. 7 2017.

[5] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. Spatiotemporal Multi-Graph Convolution Network for Ride-Hailing Demand Forecasting. Technical report, 2017.

[6] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. 5 2019.

[7] Julia Moemken, Mark Reyers, Hendrik Feldmann, and Joaquim G. Pinto. Future Changes of Wind Speed and Wind Energy Potentials in EURO-CORDEX Ensemble Simulations. *Journal of Geophysical Research: Atmospheres*, 123(12):6373–6389, 2018.

[8] Erik Delarue, Jennifer Morris, Ronald G Prinn, and John M Reilly. Renewables Intermittency: Operational Limits and Implications for Long-Term Energy System Models. *MIT Joint Program on the Science and Policy of Global Change*, (277), 2015.

[9] Kristian Larsen, Torben B Hansen, Kjeld Søballe, and Henrik Kehlet. Patient-reported outcome after fast-track hip arthroplasty: a prospective cohort study. *Health and Quality of Life Outcomes*, 8(1):144, 2010.

[10] Albert Betz. Das Maximum der theoretisch möglichen Ausnutzung des Windes durch Windmotoren. *Zeitschrift fur das gesamte Turbinenwesen 20 (1920).*, 1920.

[11] M H Albadi. Wind Turbines Capacity Factor Modeling—A Novel Approach. 24(3):1637–1638, 2009.

[12] Marcelo Gustavo Molina and Pedro Enrique Mercado. Modelling and control design of pitch-controlled variable speed wind turbines. In *Wind turbines*. In Tech, 2011.

[13] Edgar A. DeMeo, Gary A. Jordan, Clint Kalich, Jack King, Michael R. Milligan, Cliff Murley, Brett Oakleaf, and Matthew J. Schuerger. Accomodating wind's natural behavior. *IEEE Power and Energy Magazine*, 5(6):59–67, 2007.

[14] Jaesung Jung and Robert P. Broadwater. Current status and future advances for wind speed and power forecasting. *Renewable and Sustainable Energy Reviews*, 31:762–777, 2014.

[15] Kolbjørn Engeland, Marco Borga, Jean Dominique Creutin, Baptiste François, Maria Helena Ramos, and Jean Philippe Vidal. Space-time variability of climate variables and intermittent renewable electricity production – A review. *Renewable and Sustainable Energy Reviews*, 79(May 2016):600–617, 2017.

[16] Brockwell and Davis. *Introduction to Time Series and Forecasting*, volume 68. 1996.

[17] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[18] Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. pages 1–31, 2019.

[19] Kenneth B Kahn. How to measure the impact of a forecast error on an enterprise? *The Journal of Business Forecasting*, 22(1):21, 2003.

[20] Chaman L Jain. Answers to your forecasting questions. *The Journal of Business Forecasting*, 31(2):3, 2012.

[21] Rami. Krispin. *Hands-On Time Series Analysis with R : Perform Time Series Analysis and Forecasting Using R*. 2019.

[22] Rob J. Hyndman and Anne B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006.

[23] Mcelreath. *Statistical Rethinking*, volume 53. 2019.

[24] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[25] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[26] W. D. Ray, P. J. Brockwell, and R. A. Davis. *Time Series: Theory and Methods.*, volume 153. 1990.

[27] Gianluca Bontempi, Souhaib Ben Taieb, and Yann Aël Le Borgne. Machine learning strategies for time series forecasting. *Lecture Notes in Business Information Processing*, 138 LNBIP:62–77, 2013.

[28] Spyros Makridakis and Michele Hibon. ARMA models and the Box–Jenkins methodology. *Journal of Forecasting*, 16(3):147–163, 1997.

[29] Rob J Hyndman, Anne B. Koehler, J. Keith Ord, and Ralph D Snyder. *Forecasting with Exponential Smoothing*. 2008.

[30] George E P Box and David A Pierce. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association*, 65(332):1509–1526, 1970.

[31] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The M4 Competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34(4):802–808, 2018.

[32] Enish Paneru. Understanding LSTM Networks.

[33] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.

[34] Ronald J Brachman, Francesca Rossi, Peter Stone, and Series Editors. Introduction to GNNs.

[35] Raik Becker and Daniela Thrän. Completion of wind turbine data sets for wind integration studies applying random forests and k-nearest neighbors. *Applied Energy*, 208(September):252–262, 2017.

[36] Global Wind Atlas 3.0.

[37] Frede. Blaabjerg and Ke. Ma. Wind Energy Systems, reviews application of power electronics in wind energy systems. *IEEE Power and Energy Society*, 105(11):2116–2131, 2017.

[38] Miao He, Lei Yang, Junshan Zhang, and Vijay Vittal. A spatio-temporal analysis approach for short-term forecast of wind farm generation. *IEEE Transactions on Power Systems*, 29(4):1611–1622, 2014.

[39] Kedro.

appendix A

Extra Information

Some more text ...