POLITECNICO DI TORINO

Master's Degree in Data Science & Engineering

Master's Degree Thesis

Spatiotemporal Graph Neural Networks for Wind Energy Production Forecasting

Supervisors

Prof. Paolo GARZA

Dott. Luca COLOMBA

Candidate

Ivan D'ONOFRIO

December 2024

Summary

Over the past few decades, global energy consumption has steadily increased, driven by population growth, industrialization, and technological advancements. In response to the growing energy demand and the shift towards sustainable power generation, wind energy has gained significant attention due to its environmental benefits and economic viability. The main element involved in wind energy production is the wind turbine, a device that converts the kinetic energy of wind into electrical energy relying on the principle of electromagnetic induction. However, due to the inherent variability of wind patterns and environmental conditions, wind energy production is characterized by a dynamic output, which poses operational challenges for its integration into the power grid. In this context, predictive modeling of wind energy output plays a relevant role in supporting the dynamic management of wind operations, enabling strategic demand allocation and optimized use of energy storage. Recent advancements in data acquisition and control technologies have facilitated the collection and storage of large volumes of data from wind farms, which has influenced the development of sophisticated predictive models based on deep learning algorithms. The data acquired from wind farms can be naturally organized as a collection of correlated time series representing environmental and operational records for each wind turbine within a wind farm. Graph-based deep learning methods have become popular tools for processing such collections of correlated time series. Unlike traditional multivariate forecasting techniques, graph-based spatiotemporal learners leverage relational dependencies between sensors by conditioning forecasts on a (possibly dynamic) graph that spans the time series collection. This work investigates the predictive performance of Spatiotemporal Graph Neural Networks for wind energy forecasting across short and long-term predictive horizons, providing an evaluation of their effectiveness through a detailed comparative analysis against alternative architectures and highlighting their strengths and potential applications.

Acknowledgements

A Matteo, che mi ha insegnato il valore della gentilezza.
A Carmen, che mi ha insegnato il valore dell'impegno.
A Paolo, che mi ha insegnato il valore della fatica.

Table of Contents

List of Tables

List of Figures

Acronyms

ACF

Auto-Correlation Function

ARIMA

Auto-Regressive Integrated Moving Average

CNN

Convolutional Neural Network

GNN

Graph Neural Network

GPR

Gaussian Process Regression

IEA

International Energy Agency

KNN

K-Nearest Neighbors

LCOE

Levelized Cost Of Electricity

LR

Linear Regression

LSTM

Long-Short Term Memory

MAE

Mean Absolute Error

MP

Message Passing

MSE

Mean Squared Error

PCA

Principal Component Analysis

RF

Random Forest

RMSE

Root Mean Squared Error

RNN

Recurrent Neural Network

SCADA

Supervisory Control And Data Acquisition

SDWPF

Spatial Dynamic Wind Power Forecasting

STGNN

Spatio-Temporal Graph Neural Network

STMP

Spatio-Temporal Message Passing

SVM

Support Vector Machine

Chapter 1

Introduction

Over the past few decades, global energy consumption has been constantly rising, driven by population growth, industrialization, and technological advancements. According to *World Energy Outlook (2023)* [1] published by the International Energy Agency (IEA), global energy demand increased by approximately 30% between 2013 and 2023, and this trend is expected to continue in the coming years. In response to the growing energy demand and a shift towards sustainable power generation, wind energy has gained significant attention for its environmental benefits and economic viability. In 2023, wind energy accounted for nearly 7.8% of global electricity generation, up from just 2.8% in 2013 (Table 1.1). Year-on-year, the global installed wind power capacity has been increasing by an average of 10-15% over the past decade, growing from approximately 300 GW in 2013 to over 1,000 GW by the end of 2023.

Source	LCOE [$/MW h]		
	1^{st} Quartile	2^{nd} Quartile	3^{rd} Quartile
Nuclear	32.78	39.03	67.28
Wind	53.53	69.59	92.52
Gas	68.88	81.44	104.06
Hydro	67.38	90.52	113.09
Solar	62.84	95.38	126.05
Thermal	74.48	98.53	127.92
Coal	88.31	100.25	117.25
Fuel	173.12	193.76	216.30

Table 1.2: Energy sources cost comparison

Source	Supply [TW h]						
	1990	1995	2000	2005	2010	2015	2020
Coal	4.43	4.99	6.00	7.33	8.67	9.54	9.47
Gas	1.75	2.02	2.77	3.70	4.86	5.55	6.34
Hydro	2.19	2.55	2.70	3.02	3.54	3.98	4.46
Nuclear	2.01	2.33	2.59	2.77	2.76	2.57	2.68
Wind	0.00	0.01	0.03	0.10	0.34	0.83	1.60
Solar	0.00	0.00	0.00	0.00	0.03	0.24	0.83
Others	0.15	0.16	0.19	0.26	0.40	0.55	0.72
Fuel	1.32	1.22	1.18	1.13	0.96	1.02	0.67
Thermal	0.04	0.04	0.05	0.06	0.07	0.09	0.11
Total	11.89	13.32	15.51	18.37	21.63	24.37	26.88
Source	Supply share [%]						
	1990	1995	2000	2005	2010	2015	2020
Coal	37.26	37.50	38.66	39.88	40.11	39.13	35.24
Gas	14.70	15.15	17.87	20.15	22.45	22.77	23.61
Hydro	18.43	19.12	17.38	16.43	16.35	16.33	16.59
Nuclear	16.93	17.51	16.71	15.07	12.74	10.54	9.96
Wind	0.03	0.06	0.20	0.57	1.58	3.42	5.96
Solar	0.00	0.00	0.01	0.02	0.15	1.00	3.07
Others	1.27	1.17	1.20	1.43	1.84	2.24	2.68
Fuel	11.07	9.18	7.63	6.13	4.45	4.20	2.47
Thermal	0.31	0.31	0.34	0.32	0.32	0.37	0.41
Total	100.00	100.00	100.00	100.00	100.00	100.00	100.00

Table 1.1: Energy sources overview

The Levelized Cost Of Electricity (LCOE) is a measure of the average net present cost of electricity production for a generator over its lifetime, accounting for initial investment; operation; maintenance; and fuel costs:

$$\text{LCOE} = \frac{\text{sum of costs over lifetime}}{\text{sum of energy produced over lifetime}} = \frac{\sum_{y=1}^{Y} \frac{I_y+M_y+F_y}{(1+r)^y}}{\sum_{y=1}^{Y} \frac{E_y}{(1+r)^y}} \tag{1.1}$$

In year y, I_y denotes investment costs; M_y denotes operations and maintenance costs; F_y denotes fuel costs; E_y denotes the amount of electrical energy generated; r denotes the discount rate applied to energy production costs; and Y denotes the expected service life of the power station. According to the IEA, in 2020 the median wind energy production LCOE was estimated $70 per MW h, making it one of the most competitive energy sources compared to non-renewable alternatives such as coal, gas, and other fossil fuels, as summarized in Table 1.2.

This chapter provides an overview of the components and processes involved in wind energy production. Section 1.1 introduces the wind turbine, the main entity involved in wind energy generation, describing its fundamental components and role within the context of a wind farm, providing information on the mechanics, principles and physical limitations of wind energy production processes. Section 1.2 outlines the types of data that can be collected from a wind farm, describing the systems involved in the data acquisition and monitoring processes. Finally, Section 1.3 explains how the data acquired from a wind farm can be used to support wind energy management, with a focus on the role of predictive modeling in improving the operational efficiency of the wind energy infrastructure.

1.1 Wind energy production

Wind turbines and wind farms The main entity involved in wind energy production is the wind turbine, a device that converts kinetic energy of the wind into electrical power through electromagnetic induction. Figure 1.1 illustrates the main components of a modern wind turbine. The rotor, typically consisting of three blades, is the rotating part of the wind turbine that converts the wind's kinetic energy into mechanical energy. As the wind flows over the blades, it generates lift, causing the rotor to spin. The rotor is connected to the nacelle, which hosts all of the turbine's generating components. Inside the nacelle, the gearbox links the rotor's low-speed shaft to the high-speed shaft of the generator, increasing the rotation from around (15-20 rpm) to (1500-1800 rpm). The generator, also located inside the nacelle, converts the mechanical energy transmitted by the gearbox into electrical energy relying on the principle of electromagnetic induction. The controller is a digital device that manages the wind turbine's operation, monitoring system performance, and optimizing exposure to wind by adjusting the blade pitch angles and nacelle orientation. The electricity generated is transmitted through cables to the power grid or supplemental storage systems.

A group of wind turbines installed in a specific area to collectively produce electricity is known as a wind farm. By combining the effect of multiple turbines, wind farms reduce the impact of localized wind variability: if one turbine experiences lower wind speeds, others in the farm may still generate power, resulting in a more stable and predictable energy output.

Wind power equation The amount of energy a wind turbine can produce depends on design and environmental factors, including air density, the swept area of the rotor, and wind speed. Air density, influenced by altitude, temperature, and atmospheric pressure, directly affects the kinetic energy available in the wind, which defines the potential energy that a turbine can capture and convert into mechanical energy. The swept area of the rotor, instead, affects the potential volume of wind that can be intercepted by the wind turbine, making longer blades more effective at capturing energy. The power generated by a wind turbine is quantified by the wind power equation:

$$P = \frac{1}{2}\,\rho\,A\,v^3\,C_p \tag{1.2}$$

Where P [W] denotes the power output; ρ [$\mathrm{kg\,m^{-3}}$] represents air density; A [$\mathrm{m^2}$] denotes the swept area of the rotor; v [$\mathrm{m\,s^{-1}}$] denotes the wind speed; and C_p denotes the power coefficient, which represents the turbine's efficiency in converting the kinetic energy of wind into mechanical energy. The theoretical maximum efficiency of a wind turbine was established by Albert Betz in his seminal work *Wind-Energie und ihre Ausnutzung durch Windmühlen* [2]. According to Betz's Law, no wind turbine can convert more than 59.3% of the kinetic energy of the wind into mechanical energy, with modern utility-scale turbines typically achieving 75–80% of this theoretical limit.

1.2 Acquiring data from a wind farm

In modern wind farms, data collection and monitoring processes are typically conducted using a Supervisory Control And Data Acquisition (SCADA) system. SCADA is an automated control architecture that involves computers, networked data communications, and graphical user interfaces to enable centralized supervision of machines and processes. In the context of wind farms, a SCADA system

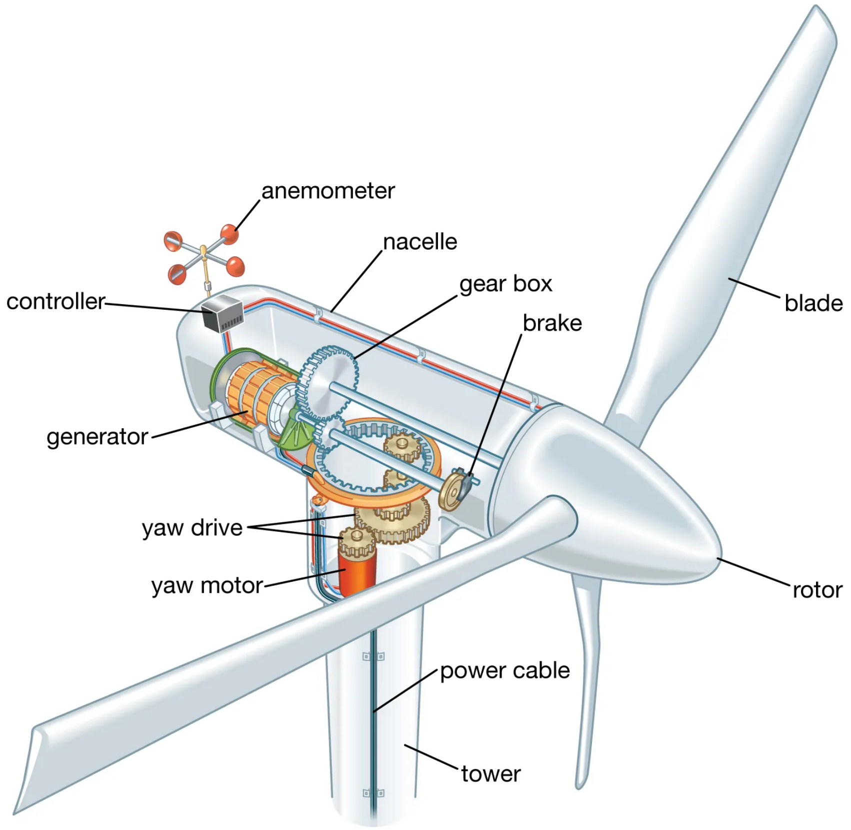


Figure 1.1: Wind turbine components

monitors each turbine's operational parameters by collecting real-time data from its sensor network, enabling remote management, performance optimization, and predictive maintenance.

Days	Interval	# columns	# channels	# turbines	# records
245	10 minutes	13	10	134	4,727,520

Table 1.3: SDWPF dataset overview

All analyses presented in this work are conducted on the Spatial Dynamic Wind Power Forecasting (SDWPF) dataset [3], which contains data recorded by the SCADA system of a wind farm operated by *Longyuan Power Group Corp. Ltd.*, the largest wind power producer in China and Asia. The dataset includes six months of records from a wind farm hosting 134 wind turbines, with data from each turbine recorded at ten-minute intervals (Table 1.3). It provides detailed information on the operational status and environmental conditions influencing wind power generation over time (Table 1.5), as well as the spatial layout of the turbines within the wind farm (Table 1.4).

1.3 Predictive modeling

Due to the inherent variability of wind patterns and environmental conditions, wind energy production is characterized by a dynamic output, which poses operational challenges for its integration into the power grid. In this context, predictive modeling of wind energy output plays a relevant role in supporting the dynamic management of wind farms by enabling strategic demand allocation [4], energy storage optimization, and the integration of complementary energy sources. Recent advancements in data acquisition and control technologies, such as SCADA (Section 1.2), have made it possible to collect and store large volumes of data from wind farms, facilitating the development of sophisticated data-driven predictive models based on machine learning algorithms.

Predictive analysis in the context of wind energy production refers to a forecasting task aiming to predict future energy output based on historical and real-time data. Depending on the application, forecasting models are used to predict energy output over different time horizons - ranging from short-term (minutes to hours) to long-term (days to weeks) - to support optimal grid management and resource allocation. This work investigates the predictive performance of Spatio-Temporal Graph Neural Network (STGNN) for wind energy forecasting (Section 2.3) across

Dimension	Units	Description
`Turb`	-	Wind turbine identifier
`Xpos`	m	Horizontal position
`Ypos`	m	Vertical position

Table 1.4: SDWPF dataset: wind turbines position

Dimension	Units	Description
`Turb`	-	Wind turbine identifier
`Day`	-	Day of the record
`Time`	-	Time of the record
`Wspd`	$\mathrm{m\,s^{-1}}$	Wind speed
`Wdir`	°	Angle between wind orientation and turbine nacelle
`Etmp`	°C	External temperature
`Itmp`	°C	Internal temperature
`Ndir`	°	Nacelle orientation
`Pab1`	°	Pitch angle of the first blade
`Pab2`	°	Pitch angle of the second blade
`Pab3`	°	Pitch angle of the third blade
`Prtv`	kW	Reactive power
`Patv`	kW	Active power

Table 1.5: SDWPF dataset: recorded channels

short and long-term predictive horizons, providing an evaluation of their effectiveness through a detailed comparative analysis (Section 4.4) against alternative approaches and highlighting their strengths and potential applications.

Chapter 2

Spatiotemporal Graph Neural Networks

Data acquired from a wind farm can be naturally organized as a collection of correlated time series representing environmental and operational parameters associated with each wind turbine. Graph-based deep learning methods, including Spatio-Temporal Graph Neural Networks (STGNN), have recently gained significant attention as tools for processing such collections of correlated time series. This chapter provides an overview of STGNNs and their applications. Section 2.1 reviews existing research on spatiotemporal forecasting. Section 2.2 formalizes the theoretical framework for spatiotemporal analysis adopted in this work. Finally, Section 2.3 introduces the most common STGNN architectures.

2.1 Related works

Existing research on wind energy production forecasting focuses on addressing challenges such as non-linear wind profiles, achieving long-term accuracy, and modeling relational dependencies between generators. Statistical approaches, such as Auto-Regressive Integrated Moving Average (ARIMA) [5, 6] and others [7, 8], have been proven effective in achieving accurate short-term forecasts due to their ability to model temporal patterns. However, due to model limitations, these methods tend to fall short in representing spatial relationships and long-term dependencies. Alternative machine learning methods, including Support Vector Machines (SVM) [9, 10], K-Nearest Neighbors (KNN) [11] and Random Forests (RF) [12], have been proven effective in representing non-linear dependencies, but lose accuracy over long-term prediction horizons. More recently, deep learning architectures, such as Convolutional Neural Networks (CNN) [13] and Recurrent Neural Networks (RNN) [14], have shown promising results in modeling long-term temporal dependencies, while Graph Neural Networks (GNN) [15, 16, 17] and Transformers [18] have been proven effective in combining temporal and spatial features, enabling modeling of relational dependencies among generators.

2.2 Theoretical framework

This section introduces a theoretical framework for spatiotemporal analysis, based on the one proposed by Cini et al. in *Graph Deep Learning for Time Series Forecasting* [19].

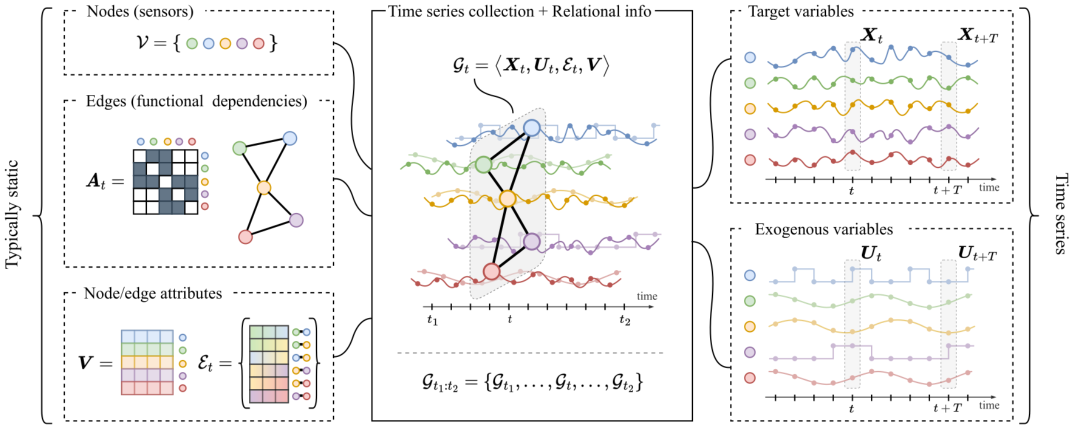

Typically static
Nodes (sensors)
$\mathcal{V} = \{ ○ ○ ○ ○ ○ \}$
Edges (functional dependencies)
$\boldsymbol{A}_t =$
Node/edge attributes
$\boldsymbol{V} =$
$\mathcal{E}_t =$
Time series collection + Relational info
$\mathcal{G}_t = \langle \boldsymbol{X}_t, \boldsymbol{U}_t, \mathcal{E}_t, \boldsymbol{V} \rangle$
time
t_1
t
t_2
$\mathcal{G}_{t_1:t_2} = \{\mathcal{G}_{t_1}, \dots, \mathcal{G}_t, \dots, \mathcal{G}_{t_2}\}$
Target variables
$\boldsymbol{X}_t$
$\boldsymbol{X}_{t+T}$
t
$t+T$
time
Exogenous variables
$\boldsymbol{U}_t$
$\boldsymbol{U}_{t+T}$
t
$t+T$
time
Time series

Figure 2.1: Theoretical framework

Consider a collection of N synchronously sampled time series, each corresponding to a sensor i recording d_x channels. Each time series is composed of a sequence of d_x-dimensional vectors $\boldsymbol{x}_t^i$ observed at each time step t. Let the matrix $\boldsymbol{X}_t \in \mathbb{R}^{N \times d_x}$ denote the stacked observable variables from all N sensors at time t; the matrix $\boldsymbol{U}_t \in \mathbb{R}^{N \times d_u}$ denote the stacked exogenous variables associated to each time series, covariate to the observables; and the matrix $\boldsymbol{V} \in \mathbb{R}^{N \times d_v}$ denote time-independent attributes associated to each sensor i. Each observation is assumed to be generated by a time-invariant stochastic process such that:

$$\boldsymbol{x}_t^i \sim p^i \left(\boldsymbol{x}_t^i \mid \boldsymbol{X}_{<t}, \boldsymbol{U}_{\leq t}, \boldsymbol{V} \right) \quad \forall i = 1, \dots, N \tag{2.1}$$

Relational dependencies among sensors are encoded as a dynamic adjacency matrix $\boldsymbol{A}_t \in \{0, 1\}^{N \times N}$ with optional edge attributes $\boldsymbol{e}_t^{ij} \in \mathbb{R}^{d_e}$ associated to each non-zero entry. The set of attributed edges encoding all available relational information is denoted by $\mathcal{E}_t = \left\{ \left\langle (i, j), \boldsymbol{e}_t^{ij} \right\rangle \mid \forall i, j : \boldsymbol{A}_t[i, j] \neq 0 \right\}$.

Scenarios where the time series recording is affected by missing data can be incorporated into this framework by introducing an auxiliary binary exogenous variable $\boldsymbol{M}_t = \{0,1\}^{N \times d_x}$, representing the availability of observations for each node and time step. Finally, the tuple $\mathcal{G}_t = \langle \boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{M}_t, \mathcal{E}_t, \boldsymbol{V} \rangle$ represents all the available information at time step t.

2.3 Introducing Spatiotemporal Graph Neural Networks

Graph-based deep learning methods have recently gained significant attention as tools for processing collections of correlated time series, enabling parameter sharing during the processing of time series while conditioning predictions on the information carried by neighboring entities. GNNs are generally designed to constrain the information flow in the network relying on a graphical representation of the data. The most common GNN architectures, known as spatial GNNs, rely on the Message Passing (MP) operation [20].

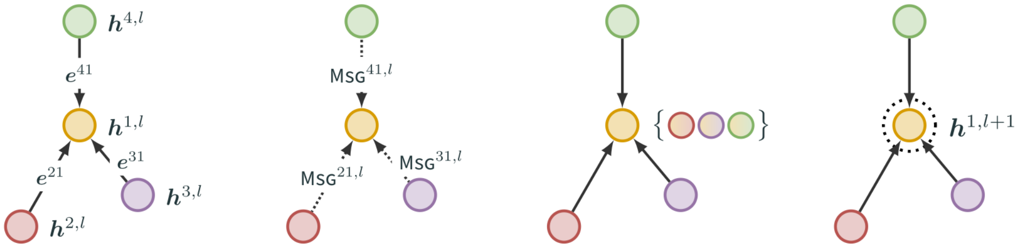


Figure 2.2: Message passing operation

Given a graph with static node features and edge set $\mathcal{E}$, MP neural networks are built by stacking MP layers that iteratively update each node's representation. The message-passing operation for node i at layer l is defined as:

$$\boldsymbol{h}^{i,l+1} = \mathsf{Up}^l\left(\boldsymbol{h}^{i,l}, \mathsf{Aggr}\left(\left\{\mathsf{Msg}^l\left(\boldsymbol{h}^{i,l}, \boldsymbol{h}^{j,l}, \boldsymbol{e}^{ij}\right)\right\}_{j \in \mathcal{N}(i)}\right)\right) \tag{2.2}$$

Where $\mathsf{Up}(\cdot)$ denotes the update function; $\mathsf{Msg}(\cdot)$ denotes the message function; and $\mathsf{Aggr}(\cdot)$ denotes a permutation-invariant aggregation function applied over the set of neighbors $\mathcal{N}(i)$. Spatio-Temporal Graph Neural Networks (STGNN) [19] [21] can be designed by extending the MP operation to handle temporal information. The resulting Spatio-Temporal Message Passing (STMP) operation aggregates at each time frame sequences of historical information from the neighborhood of each node i and it is defined as:

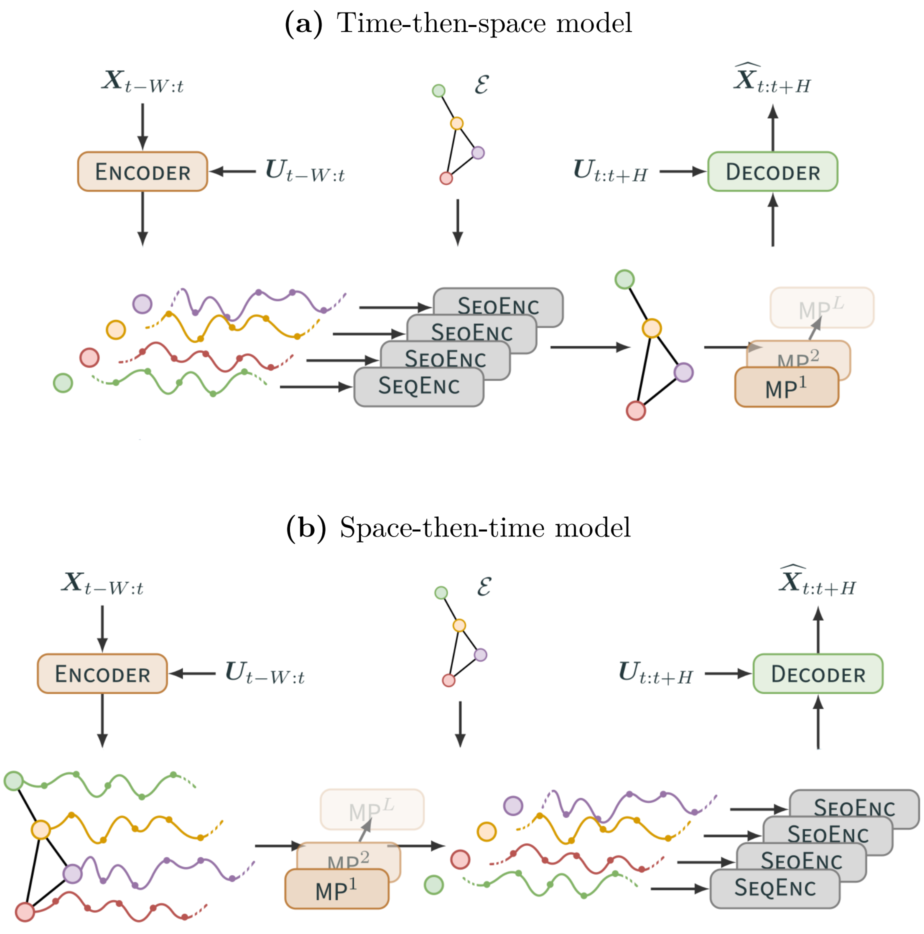
$$\boldsymbol{h}_t^{i,l+1} = \mathsf{Up}^l\left(\boldsymbol{h}_{\leq t}^{i,l}, \mathsf{Aggr}\left(\left\{\mathsf{Msg}^l\left(\boldsymbol{h}_{\leq t}^{i,l}, \boldsymbol{h}_{\leq t}^{j,l}, \boldsymbol{e}_{\leq t}^{ij}\right)\right\}_{j\in\mathcal{N}_t(i)}\right)\right) \tag{2.3}$$

where $\mathcal{N}_t(i)$ represents the set of neighbors of node i at time t and $\boldsymbol{h}_{\leq t}^{i,l}$ denotes node representation up to time t. Following the notation proposed by Cini et al., predictive modeling using STGNN can be described as a sequence of three operations:

$$\begin{aligned}
&\boldsymbol{h}_{t-1}^{i,0} = \mathsf{Encoder}\left(\boldsymbol{x}_{t-1}^i, \boldsymbol{u}_{t-1}^i, \boldsymbol{v}^i\right)\\
&\boldsymbol{H}_{t-1}^{l+1} = \mathsf{STMP}\left(\boldsymbol{H}_{\leq t-1}^l, \mathcal{E}_{\leq t-1}\right) \quad \forall l = 0, \ldots, L-1\\
&\boldsymbol{y}_t^i = \mathsf{Decoder}\left(\boldsymbol{h}_{t-1}^{i,L}, \boldsymbol{u}_t^i\right)
\end{aligned} \tag{2.4}$$

where $\mathsf{Encoder}(\,\cdot\,)$ and $\mathsf{Decoder}(\,\cdot\,)$ denote generic input and output layers, respectively, which do not propagate spatiotemporal information; $\boldsymbol{H}_t$ denotes a STGNN operation on all the nodes; and $\boldsymbol{y}_t^i$ denotes the prediction target.

Adopting the same taxonomy introduced in [19], STGNN are categorized based on whether the temporal and spatial dimensions can be factorized into separate steps and the order in which spatial and temporal operations are applied. Time-then-space models (Fig. 2.3a) first process the sequence of node representations $\boldsymbol{h}_{<t}^{i,0}$ using a temporal model before applying any spatial message-passing operations. Space-then-time models (Fig. 2.3b), instead, apply spatial message-passing first, followed by temporal processing. Time-and-space models, instead, simultaneously process temporal and spatial information in an integrated framework.


(a) Time-then-space model

(b) Space-then-time model

Figure 2.3: STGNN models

Chapter 3

Data preparation and exploration

This chapter focuses on the the exploration of the SDWPF dataset. Section 3.1 introduces the types of outliers present in the dataset and the methodologies used for their identification. Section 3.2 describes the approach adopted to reconstruct unavailable records, including the application of spatial interpolation techniques based on Gaussian Process Regression. Finally, Section 3.3 provides an overview of the dataset's spatial and temporal characteristics, focusing on the wind farm's spatial layout, the distribution of recorded channels, their pairwise correlations, and temporal autocorrelation profiles.

3.1 Data preparation

The SDWPF dataset contains two distinct types of outliers caused by issues in the data acquisition process of individual wind turbines. Operational outliers result from lapses in the SCADA system's recording process, where failures compromise the integrity of all monitored channels over specific time frames. On the other hand, readout outliers are due to anomalies in individual sensors, causing observed values to significantly deviate from the expected operational ranges of a given channel.

3.1.1 Operational outliers

The criteria for identifying operational outliers are outlined in [3], which defines the boundaries for nominal wind turbines operational ranges and specifies conditions for detecting records that are incongruent with expected grid operations or environmental physics, including anomalies in recorded temperatures, wind profiles, and turbine mechanical configurations. To identify operational outliers, these conditions are translated into a set of logical statements (reported in Table 3.1) and evaluated using a custom naïve rule-based estimator. The estimator tests each dataset entry against all logical statements, flagging the ones that meet at least one condition as operational outlier. This information is then encoded as an additional `Mask` channel and incorporated into the dataset.

Analyzing the distribution of operational outliers across the wind farm (Figure 3.1), it is possible to identify which wind turbines are affected by severe data integrity issues. Defining a threshold on the maximum acceptable proportion of operational outliers - corresponding to 40% of each turbine's total records - $\texttt{Turb}_{24}$, $\texttt{Turb}_{25}$, $\texttt{Turb}_{38}$, $\texttt{Turb}_{61}$, $\texttt{Turb}_{67}$, $\texttt{Turb}_{68}$, $\texttt{Turb}_{121}$, and $\texttt{Turb}_{122}$ are classified unsuitable for further analysis and excluded from the dataset.

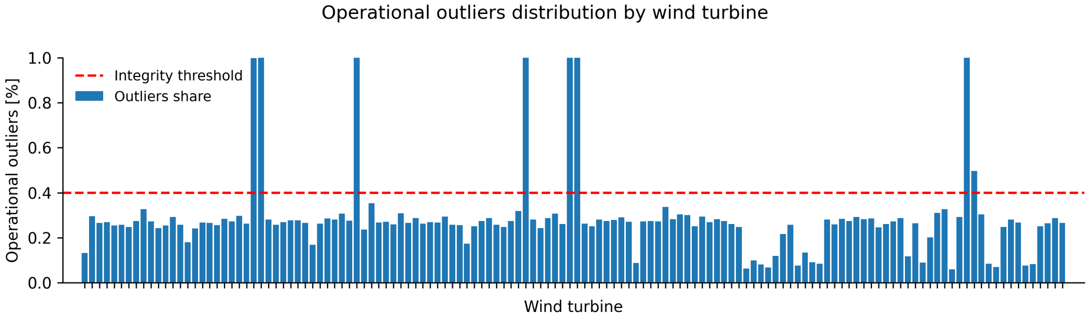


Figure 3.1: Operational outliers distribution across wind turbines

Figure 3.2 shows a comparative analysis of the channels recoded by $\texttt{Turb}_1$ before and after the identification of operational outliers. The first chart (Figure 3.2a) contains all the dataset entries associated to $\texttt{Turb}_1$; while the second chart (Figure 3.2b) includes only the valid entries, with operational outliers excluded. Unavailable records are marked in red relying on the newly introduced `Mask` channel.

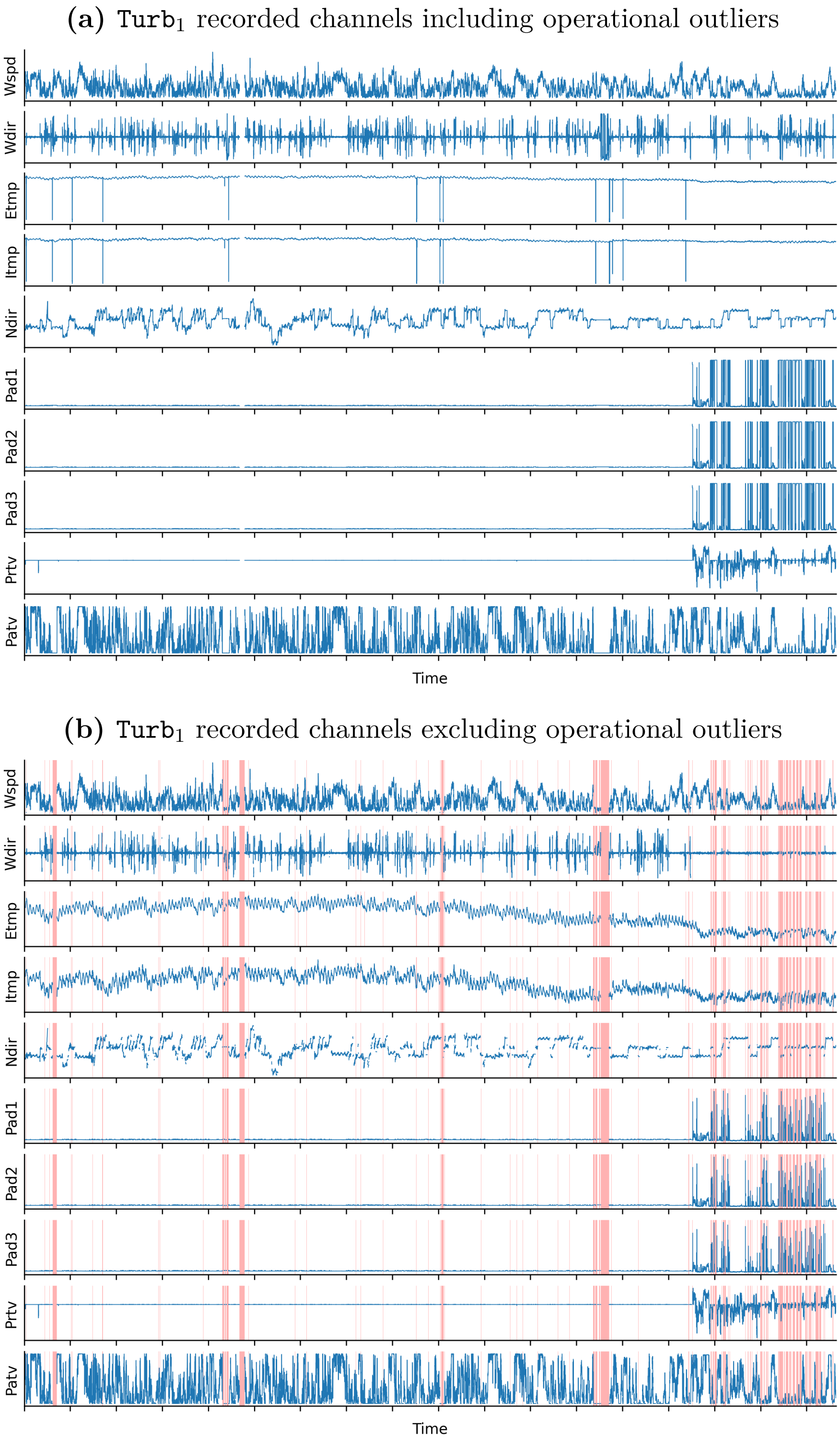
3.1.2 Readout outliers

The methodology applied for identifying readout outliers relies on time series seasonal decomposition. The collection of historical records from a wind turbine channel is represented as a time series $S = \{S_t\}_{t=1}^T$ of length T. A time series can be decomposed into three additive signals (Figure 3.3):

$$S = \Theta + \Sigma + \epsilon \tag{3.1}$$

Rule	Condition
	Active power
01	$\texttt{Patv} < 0\text{kW}$
02	$\texttt{Patv} = 0\text{kW} \wedge \texttt{Wspd} > 2.5\,\text{m}\,\text{s}^{-1}$
	Wind speed
03	$\texttt{Wspd} < 1\,\text{m}\,\text{s}^{-1} \wedge \texttt{Patv} > 10\text{kW}$
04	$\texttt{Wspd} < 2\,\text{m}\,\text{s}^{-1} \wedge \texttt{Patv} > 100\text{kW}$
05	$\texttt{Wspd} < 3\,\text{m}\,\text{s}^{-1} \wedge \texttt{Patv} > 200\text{kW}$
	Wind direction
06	$\texttt{Wdir} < -180°$
07	$\texttt{Wdir} > 180°$
	Temperature
08	$\texttt{Etmp} < -21°\text{C}$
09	$\texttt{Etmp} > 60°\text{C}$
10	$\texttt{Itmp} < -21°\text{C}$
11	$\texttt{Itmp} > 70°\text{C}$
	Mechanical configuration
12	$\texttt{Ndir} < -720°$
13	$\texttt{Ndir} > 720°$
14	$\texttt{Pab1} > 89°$
15	$\texttt{Pab2} > 89°$
16	$\texttt{Pab3} > 89°$
	Missing records
17	$\texttt{Wspd} = 0\,\text{m}\,\text{s}^{-1} \wedge \texttt{Wdir} = 0° \wedge \texttt{Etmp} = 0°\text{C}$

Table 3.1: Operational outliers rule-based system



(a) `Turb`$_1$ recorded channels including operational outliers

(b) `Turb`$_1$ recorded channels excluding operational outliers

Figure 3.2: Operational outliers comparative analysis

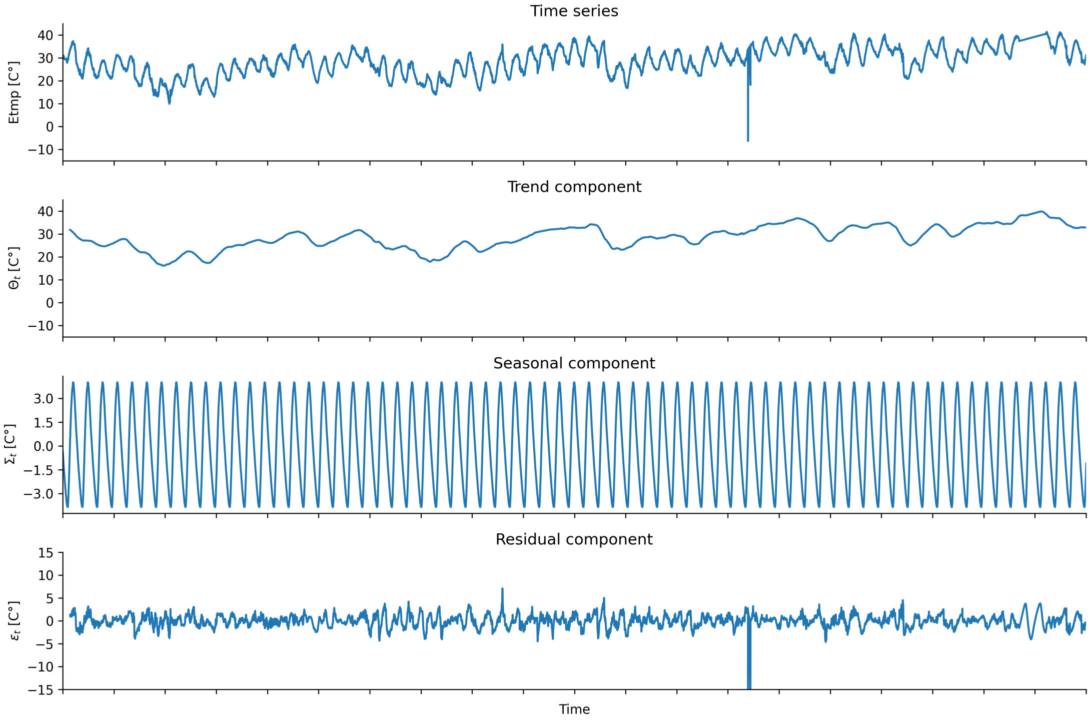


Figure 3.3: Time series seasonal decomposition

Trend component The trend component $\Theta = \{\Theta_t\}_{t=1}^{T}$ describes the overall trajectory of the time series. It is computed as the moving average over a window that spans multiple seasonal cycles. Given a seasonal period p, the trend value Θ_t at time t is defined as:

$$\Theta_t = \frac{1}{p} \sum_{i=-p/2}^{p/2} S_{t+i} \tag{3.2}$$

Seasonal component The seasonal component $\Sigma = \{\Sigma_t\}_{t=1}^{T}$ describes periodic fluctuations within time series. It is computed by averaging the detrended series values that occur at the same position within each seasonal cycle. Let $C = T \mod p$ represent the number of complete seasonal cycles in the time series. The seasonal value Σ_t at time t is defined as:

$$\Sigma_t = \frac{1}{C} \sum_{c=1}^{C} (S_{t+cp} - \Theta_{t+cp}) \tag{3.3}$$

Residual component The residual component $\epsilon = \{\epsilon_t\}_{t=1}^{T}$ describes the deviation from the combined trend and seasonal components. It is computed by removing the estimated trend and seasonal components from the original time series. The residual value ϵ_t at time t is defined as:

$$\epsilon_t = S_t - \Theta_t - \Sigma_t \tag{3.4}$$

An observation t is labeled as readout outlier if its corresponding residual ϵ_t diverges from the expected residuals' distribution by a magnitude greater than δ standard deviations σ_ϵ:

$$|\epsilon_t| > \delta \, \sigma_\epsilon \tag{3.5}$$

where δ denotes a user-defined sensitivity parameter and σ_ϵ denotes the standard deviation of the residual time series ϵ. The outliers identified following this procedure have been flagged and then reconstructed as described in Section 3.2. Figure 3.4 shows the result of the outlier detection routine on the `Etmp` channel of wind turbine $\texttt{Turb}_1$ with $p = 144$ (one day) and $\delta = 3$.

3.2 Data imputation

This section describes the imputation strategy used to reconstruct the `Etmp` and `Itmp` channels for wind turbines affected by readout outliers. The adopted methodology applies Gaussian Process Regression (GPR or Kriging) to interpolate unknown

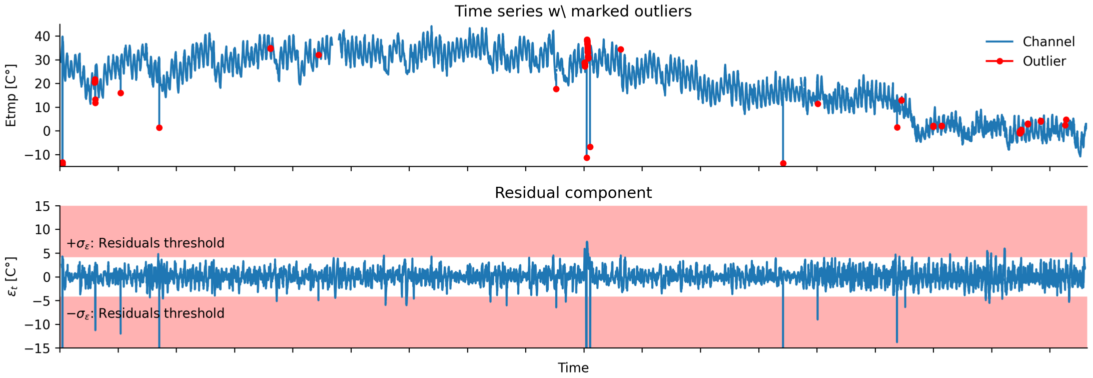


Figure 3.4: Outlier detection on the `Etmp` channel recorded by `Turb`$_1$

temperature values relying on spatial correlations among wind turbines. Essentially, this approach consumes the temperature data recorded by operational wind turbines in a specific time frame t to produce a spatial temperature model (Figure 3.5). The trained model is then applied to estimate temperature values for the remaining non-operational wind turbines.

3.2.1 Spatial interpolation

Kriging is a spatial interpolation method used to make predictions at unsampled locations based on observed geostatistical data. Given a set of observed data points $\mathcal{Z} = \{Z(\boldsymbol{l}_i)\}_{i=1}^{N}$, the objective is to estimate the value of Z at an arbitrary unsampled location $\boldsymbol{l}_0$. The Kriging estimator, $\hat{Z}\left(\boldsymbol{l}_0\right)$, is defined as a linear, unbiased estimator:

$$\hat{Z}\left(\boldsymbol{l}_0\right) = \sum_{i=1}^{N} \lambda_i Z\left(\boldsymbol{l}_i\right) \quad s.t. \quad \sum_{i=1}^{N} \lambda_i = 1 \tag{3.6}$$

where λ_i are weights chosen to minimize the mean squared prediction error:

$$\mathbb{E}\left[\left(\hat{Z}\left(\boldsymbol{l}_0\right) - Z\left(\boldsymbol{l}_0\right)\right)^2\right] \tag{3.7}$$

The Kriging estimator relies on the definition of a variogram $\gamma\left(h\right)$ which quantifies spatial correlation between data points as a function of their separation distance $h = \|\boldsymbol{l}_i - \boldsymbol{l}_j\|$:

$$\gamma\left(h\right) = \frac{1}{2}\,\mathbb{E}\left[\left(Z(\boldsymbol{l}) - Z(\boldsymbol{l} + h)\right)^2\right] \tag{3.8}$$

The weights λ_i are obtained by solving the Kriging system of $N + 1$ linear equations given by:

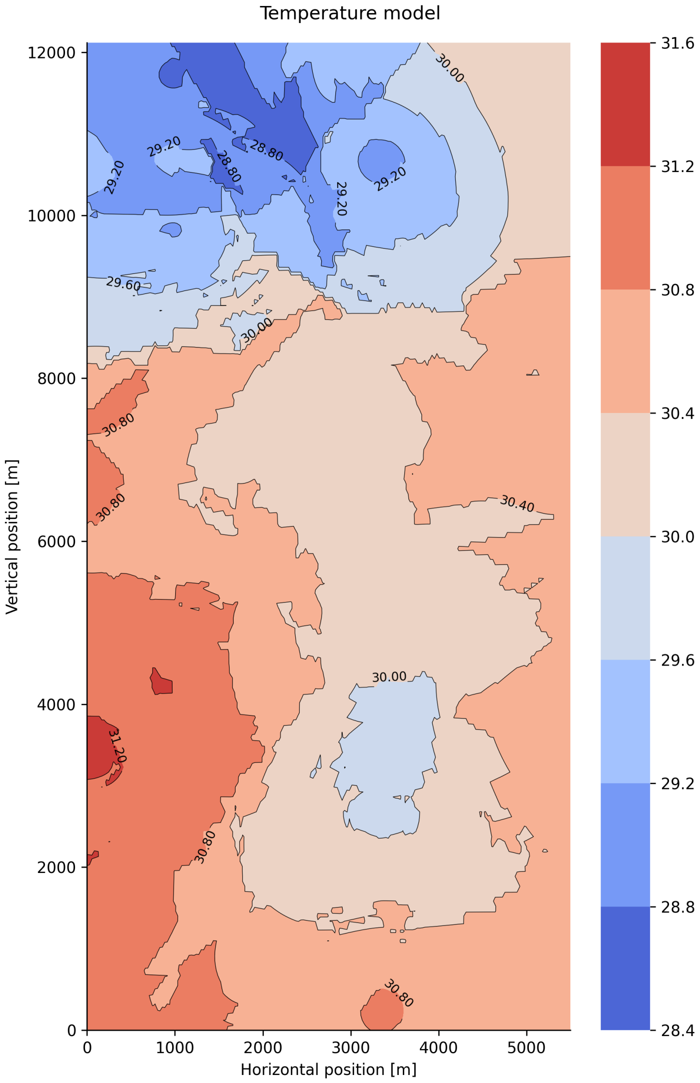


Figure 3.5: Temperature model

$$\sum_{j=1}^{N} \lambda_j \gamma\left(\|\boldsymbol{l}_i - \boldsymbol{l}_j\|\right) + \mu = \gamma\left(\|\boldsymbol{l}_i - \boldsymbol{l}_0\|\right) \quad \forall i = 1, \ldots, N \tag{3.9}$$

3.2.2 Variogram model selection

In Kriging, the variogram model defines the spatial correlation structure of the data. Each model is characterized by specific parameters that describe the spatial relationship between observations. The nugget κ_0 denotes measurement error at very short distances, setting the variogram's initial value as distance approaches zero. The sill κ denotes the maximum variance of the variogram, quantifying the limit of spatial correlation. In finite models, the range R denotes the maximum distance where correlation is significant. In continuous models, instead, the decay parameter α denotes how quickly correlation decreases with distance.

Spherical model The spherical model (Figure 3.6a) describes a spatial relationship characterized by correlation increasing with distance up to a maximum range R, beyond which it becomes negligible:

$$\gamma(h) = \begin{cases} \kappa_0 + \kappa\left(\frac{3h}{2R} - \frac{h^3}{2R^3}\right) & \text{if } h \leq R \\ \kappa_0 + \kappa & \text{if } h > R \end{cases} \tag{3.10}$$

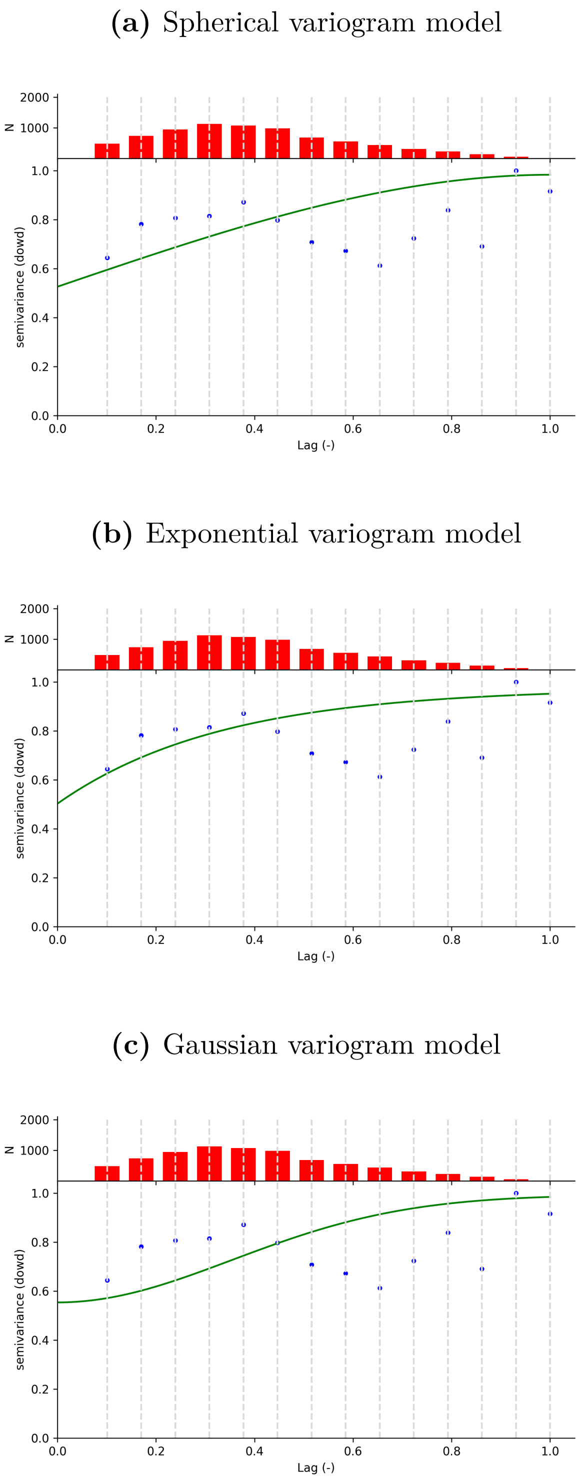
Exponential model The exponential model (Figure 3.6b) describes a spatial relationship characterized by correlation asymptotically approaching zero:

$$\gamma(h) = \kappa_0 + \kappa\left(1 - e^{-\frac{h}{\alpha}}\right) \tag{3.11}$$

Gaussian model The Gaussian model (Figure 3.6c) describes a spatial relationship characterized by correlation approaching zero following a normal profile:

$$\gamma(h) = \kappa_0 + \kappa\left(1 - e^{-\frac{h^2}{\alpha^2}}\right) \tag{3.12}$$

To determine the best variogram model for temperature estimation, this analysis compared the spherical, exponential, and Gaussian models, evaluating each by the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) between the empirical variogram and the model predictions (Table 3.2). Based on these results, the Gaussian model was selected as the best suitable for the interpolation process.



(a) Spherical variogram model

(b) Exponential variogram model

(c) Gaussian variogram model

Figure 3.6: Variogram models

Model	MAE		RMSE	
	Avg.	Std.	Avg.	Std.
Spherical	0.00011	0.00015	0.00033	0.00043
Exponential	0.00011	0.00015	0.00034	0.00045
Gaussian	0.00011	0.00014	0.00032	0.00042

Table 3.2: Variogram model selection

Figure 3.7 shows the result of the outlier interpolation routine on the `Etmp` channel of wind turbine `Turb`$_1$.

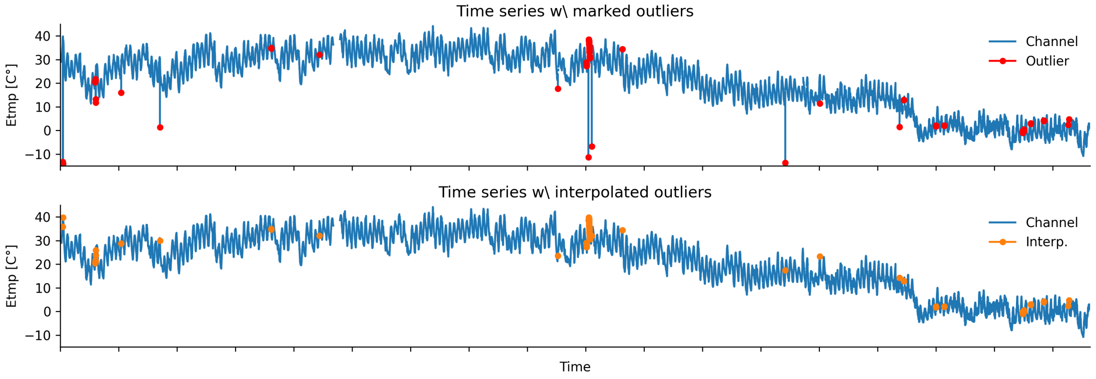


Figure 3.7: Outlier interpolation on the `Etmp` channel recorded by `Turb`$_1$

3.3 Data exploration

This section explores the spatial and temporal characteristics of the dataset, focusing on turbine positions, channel distributions, correlations, and temporal dependencies.

3.3.1 Wind turbines position

The wind farm under analysis consists of 134 wind turbines distributed over an area of 5.5km × 12km (Figure 3.8). The SDWPF dataset reports each turbine's position (Table 1.4) as an absolute horizontal (`Xpos`) and vertical (`Ypos`) distance from an arbitrary origin. While information on the farm's precise location and specific environmental characteristics (e.g., altitude, terrain features) is not available, the turbine layout shows distinct spatial clusters separated by larger gaps (Figure 3.9). The analysis of this arrangement is beyond the scope of this study.

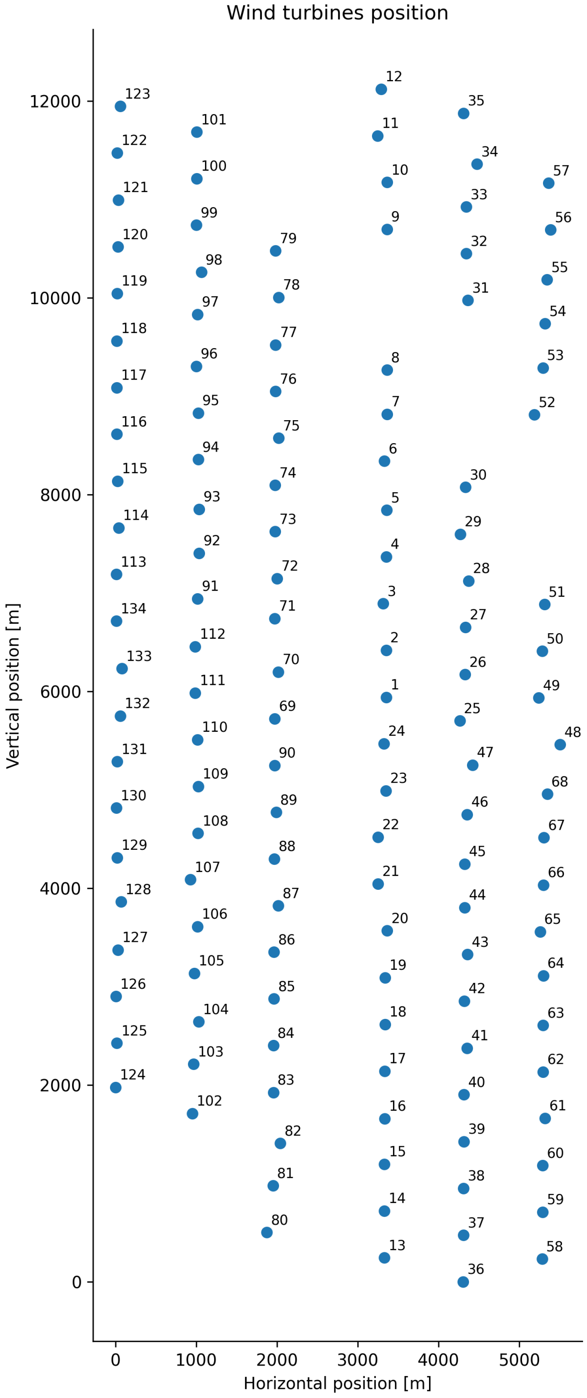


Figure 3.8: Wind turbines position

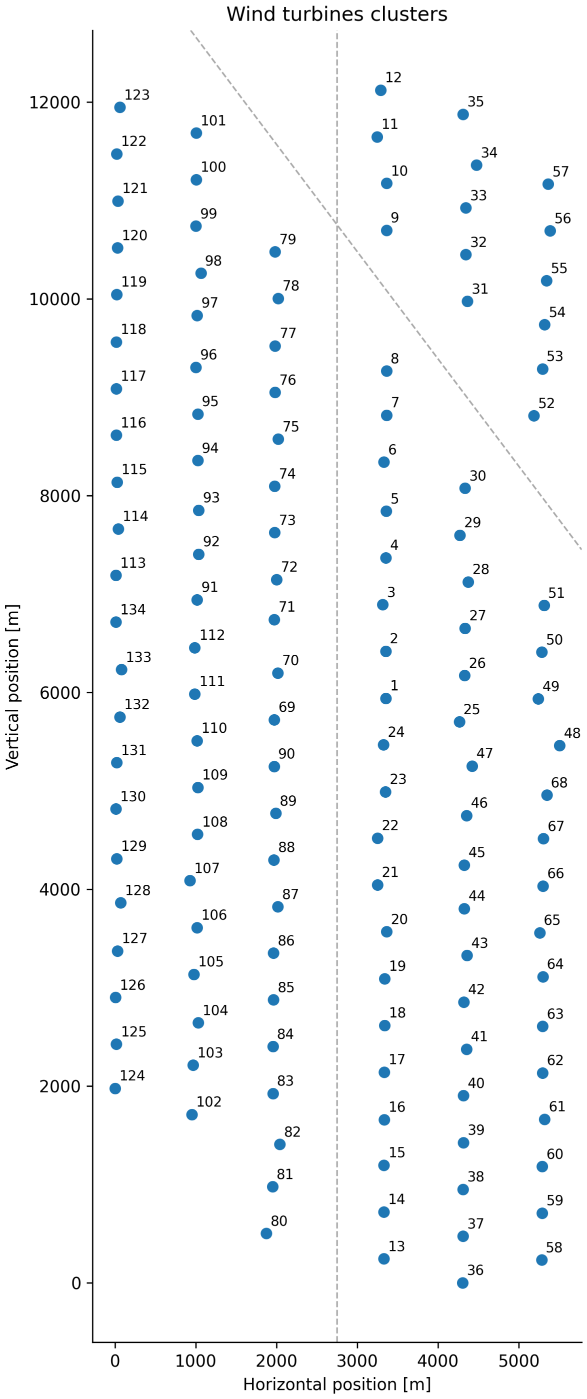


Figure 3.9: Wind turbines clusters

3.3.2 Channels distribution analysis

Distribution analysis assesses typical operating conditions and distributional patterns observed within the data. Figure 3.10 presents the statistical distribution of each channel recorded in the SDWPF dataset.

Wind and nacelle direction The `Wdir` channel, measuring the angle between the wind direction and the turbine nacelle's axis, shows a distribution that peaks at 0°. This pattern reflects the turbines' automatic alignment with the wind direction to optimize exposure and maximize energy production. Similarly, the `Ndir` channel -which measures nacelle's absolute orientation - shows a circular pattern, representing the movement described by the wind turbine.

Wind speed and active power The `Patv` and `Wspd` channels, measuring active power generation and wind speed respectively, show similar distributions. Unlike `Wspd`, `Patv` shows two peaks: one at 0 kW, representing non-operational conditions, and another around 1500 kW, corresponding to the maximum output of the turbine generator. Beyond this threshold, power output plateaus due to turbine's physical constraints.

Reactive power The `Prtv` channel, measuring reactive power output, shows frequent negative values. Negative reactive power indicates power flowing from the grid to the generator, typically occurring when the generator is under-excited. This condition reduces stability and increases the risk of the generator losing synchronism with the system, potentially causing mechanical and electrical damage. In this context, the reactive power flow stabilizes the system and prevents disruptions in the power grid.

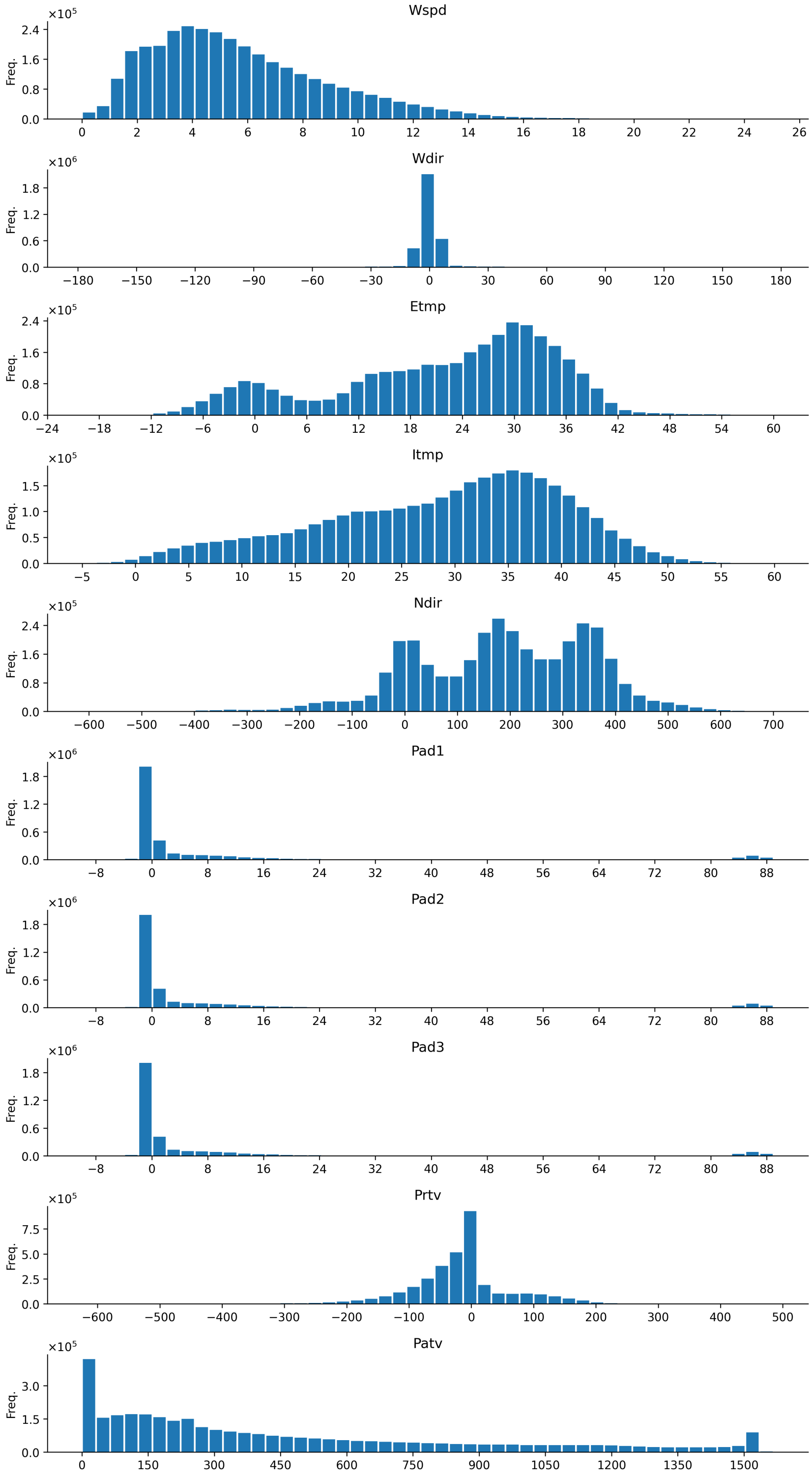
3.3.3 Channels correlation analysis

Correlation analysis aims at evaluating linear relationships occurring between time series. This analysis relies on the Pearson's correlation coefficient, which quantifies the strength and the direction of linear relationships between two time series S_a and S_b. The correlation coefficient $\rho \in [-1, 1]$ is defined as:

$$\rho = \frac{\mathrm{Cov}\,(S_a, S_b)}{\sigma_{S_a}\ \sigma_{S_b}} \tag{3.13}$$

Where $\mathrm{Cov}\,(S_a, S_b)$ denotes the covariance between the two time series, while σ denotes their standard deviations. A coefficient $\rho \sim \pm 1$ indicates a perfect linear correlation - positive or negative, respectively - while $\rho \sim 0$ indicates the absence

Figure 3.10: Channels distribution



of any linear relationship. Figure 3.11 represents the pairwise Pearson's correlation matrix between all the channels recorded in the SDWPF dataset.

Blades pitch angles The `Pab1`, `Pab2` and `Pab3` channels, measuring the pitch angles of the first, second, and third wind turbine blades respectively, are perfectly correlated. These angles are adjusted simultaneously, allowing turbines to optimize the blades' angle of attack to maximize energy production or halt power generation.

Internal and external temperatures The `Etmp` and `Itmp` channels, measuring the temperature of the surrounding environment and inside the wind turbine nacelle respectively, are strongly but not perfectly correlated. Unlike `Etmp`, which only describes the external temperature, `Itmp` records the combined effect of both, the external temperature and the heat produced by internal nacelle components, such as the generator. Monitoring `Itmp` ensures that wind turbine's internal components operate within operational limits.

Wind speed and active power The `Patv` and `Wspd` channels, measuring active power output and wind speed, respectively, are strongly but not perfectly correlated. Active power output depends non-linearly on wind speed (Eq. 1.2), and it's influenced by external factors such as air pressure, temperature, wind turbine's orientation relative to the wind and additional operational conditions. These elements combined add complexity to the relationship between wind speed and power generation, which cannot be fully captured by the linear correlation coefficient.

3.3.4 Channels temporal autocorrelation analysis

Temporal autocorrelation analysis aims at identifying information persistency over time in time series. This analysis relies on the Auto-Correlation Function (ACF), which quantifies correlation between observations at various time lags within a time series S. A time lag k represents the interval between the current value and a prior value occurred k time frames earlier. The autocorrelation coefficient $\rho_k \in [-1, 1]$ at lag k is defined as:

$$\rho_k = \frac{\mathrm{Cov}\left(S, S_k\right)}{\sigma_S^2} \tag{3.14}$$

Where $\mathrm{Cov}\left(S, S_k\right)$ denotes the covariance between the current and lagged values, while σ_S^2 denotes the variance of S. A coefficient $\rho_k \sim \pm 1$ indicates information persistence over time, while $\rho_k \sim 0$ indicates the absence of any temporal relationship. Figure 3.12 shows the correlogram for each channel recorded in the SDWPF

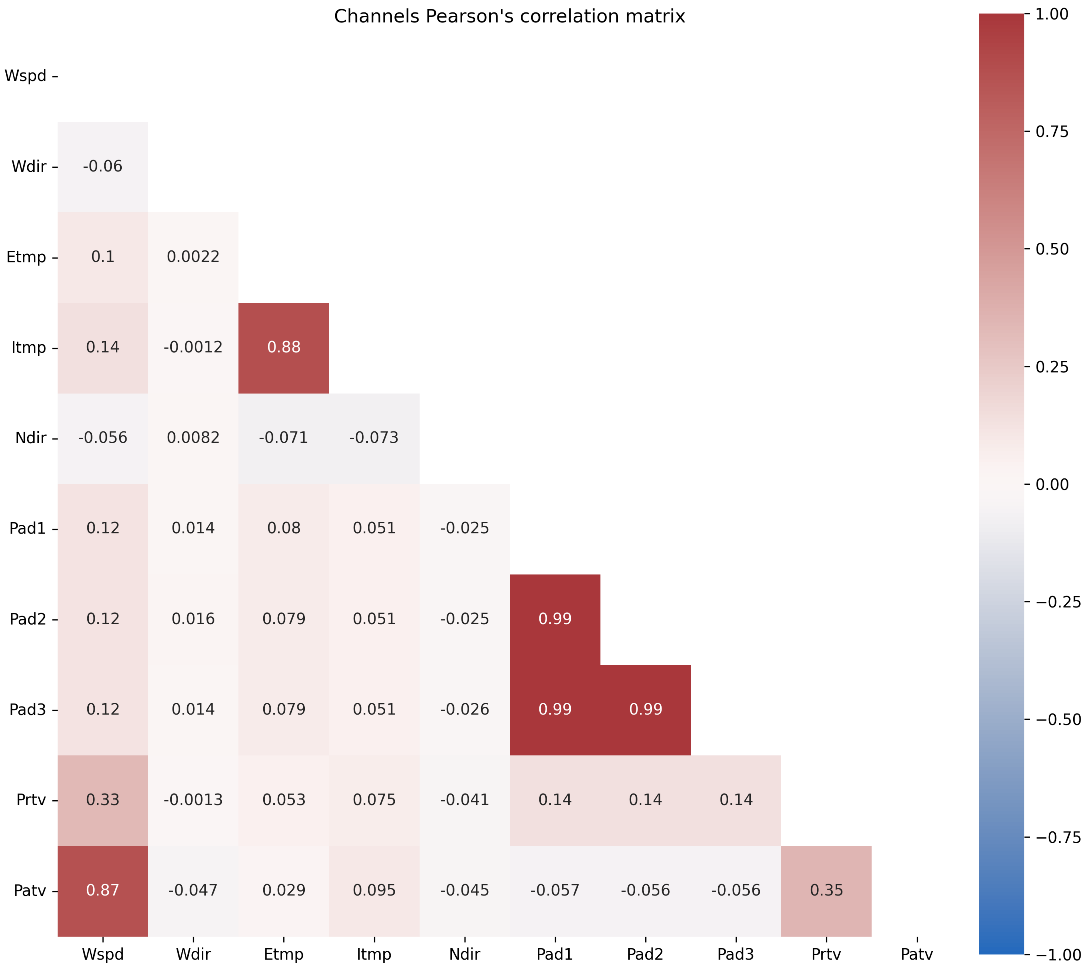


Figure 3.11: Channels Pearson's correlation matrix

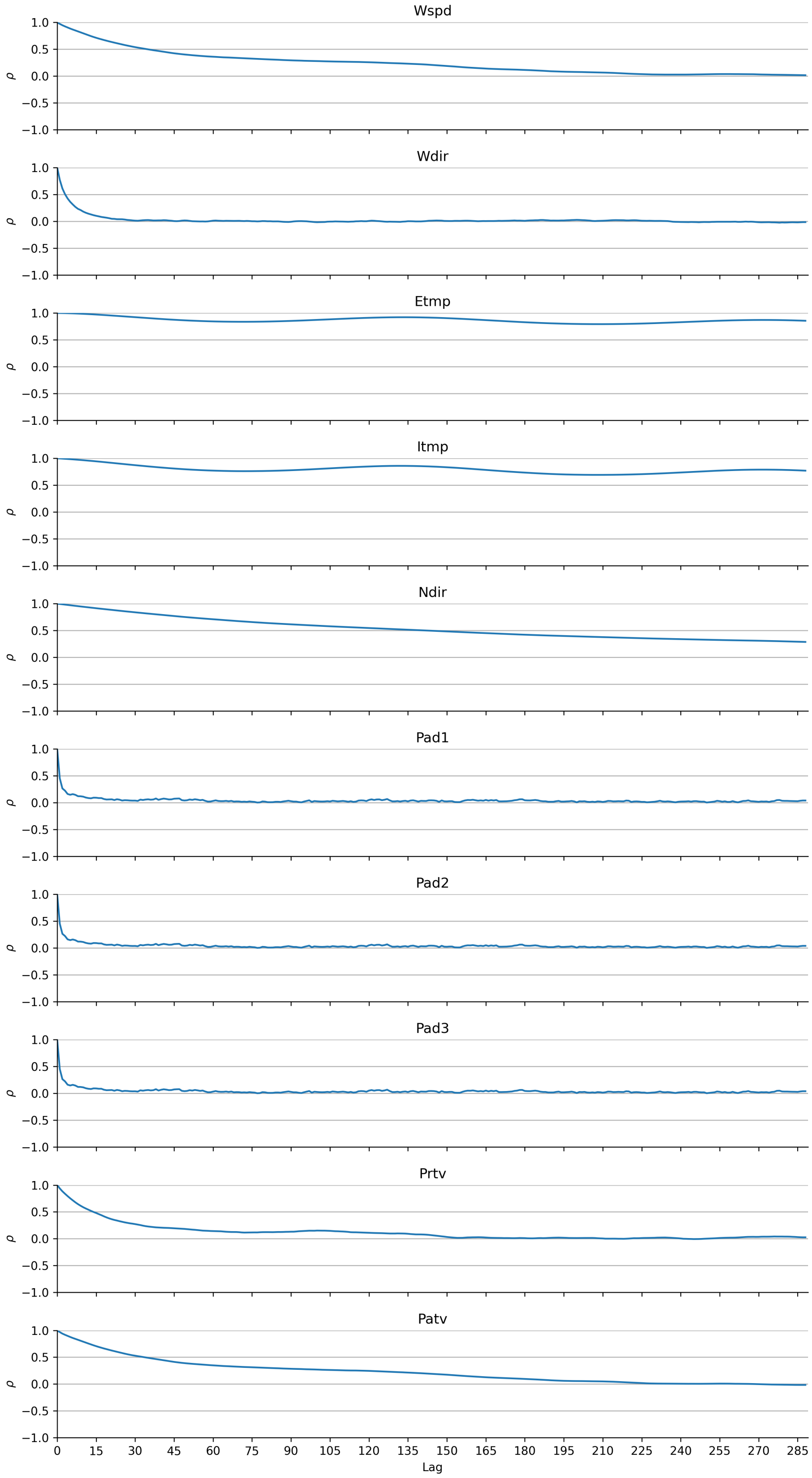
dataset over a two days period assuming an autocorrelation significance threshold $\rho^* = 0.5$.

Internal and external temperature Both the `Etmp` and `Itmp` channels show similar autocorrelation patterns, with peaks every 144 lags, corresponding to a 24 hours period. This suggests a daily temperature cycle, where temperatures at the same hour on consecutive days are likely to be similar.

Wind speed and active power The `Wspd` and `Patv` channels show similar autocorrelation patterns. With values falling below ρ^* after approximately 40 lags, corresponding to a 6 hours period. This suggests that only the most recent 6 hours of data are informative for understanding variations in wind speed and, consequently, energy production.

Wind direction The `Wdir` channel shows a rapid decrease in autocorrelation, falling below ρ^* after 5 lags, approximately corresponding to 1 hour. This behavior indicates high variability in wind direction over short time intervals.

Figure 3.12: Channels temporal autocorrelation



Chapter 4

Experimental setting

This chapter presents the experimental framework adopted to investigates the predictive performance of Spatio-Temporal Graph Neural Networks for wind energy forecasting across short and long-term predictive horizons. Section 4.1 introduces the multi-step ahead wind energy production task, describing how the data recorded in the SDWPF dataset can be integrated in the theoretical framework for spatiotemporal analysis presented in Section 2.2. Section 4.2 describes the methodological framework used for the training and evaluation of the models under analysis. Section 4.3 outlines the different machine learning architectures involved in the experiments and their configuration. Finally, Section 4.4 provides an evaluation of the effectiveness of STGNNs through a comparative analysis against alternative architectures, highlighting their strengths and potential applications.

4.1 Wind energy production forecasting

This section describes how the data recorded in the SDWPF dataset can be integrated in the theoretical framework for spatiotemporal analysis presented in Section 2.2 and introduces the resulting wind energy production multi-step ahead forecasting problem.

4.1.1 Framework integration

The records associated to the $N = 126$ viable wind turbines (Section 3.1.1) are aligned with the theoretical framework's (Section 2.2) entities. The active power output, serving as the primary observable variable and target of the analysis, is stored in $\boldsymbol{X}_t$. The exogenous variables - including the temporal dimension; wind speed and direction; external and internal temperatures; nacelle orientation; blade pitch angles; and reactive power - are stored in $\boldsymbol{U}_t$. The observations availability indicator - denoted by the mask introduced in Section 3.1.1 - is stored in $\boldsymbol{M}_t$. In this context, the observable variable is one-dimensional, and both the observation $\boldsymbol{x}_t^i$ and the mask $\boldsymbol{m}_t^i$ simplify to scalar values (i.e., since $d_x = 1$, then $\boldsymbol{X}_t = \{x_t^i\}_{i=1}^N$ and $\boldsymbol{M}_t = \{m_t^i\}_{i=1}^N$).

4.1.2 Multi-step ahead forecasting

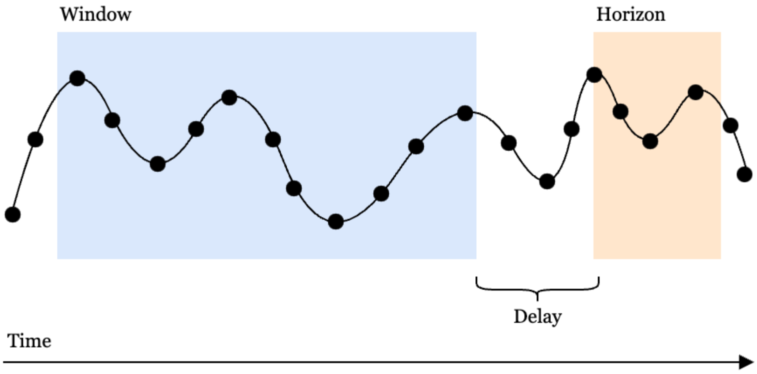


Figure 4.1: Window and horizon

The prediction task is formulated as a multi-step ahead forecasting problem. The goal of this exercise is to learn a model $p_{\boldsymbol{\theta}}$ parametrized by $\boldsymbol{\theta}$ able to predict future observations over a forecasting horizon $H \geq 1$ relying on a backward-looking window $W \geq 1$ of historical observations, approximating the unknown conditional probability distribution:

$$\begin{aligned} p_{\boldsymbol{\theta}} \left(\boldsymbol{x}_{t:t+h}^i \mid \boldsymbol{X}_{t-W:t}, \boldsymbol{U}_{t-W:t+h+1}, \boldsymbol{V} \right) \approx \\ p^i \left(\boldsymbol{x}_{t:t+h}^i \mid \boldsymbol{X}_{<t}, \boldsymbol{U}_{\leq t+h+1}, \boldsymbol{V} \right) \\ \forall h \in [0, H) \,; \, \forall i = 1, \dots, N \end{aligned} \tag{4.1}$$

In this specific application, the exogenous variables $\boldsymbol{U}$, which are covariates to the target, are available up to time $t-1$. The model can be rewritten as:

$$p_{\boldsymbol{\theta}} \left(\boldsymbol{x}_{t:t+h}^i \mid \boldsymbol{X}_{t-W:t}, \boldsymbol{U}_{t-W:t}, \boldsymbol{V} \right) \tag{4.2}$$

To account for the relational dependencies among wind turbines, the model is conditioned by incorporating relational information represented in the form of a relational graph $\mathcal{E}_t$. The conditioning on $\mathcal{E}_t$ biases the model into accounting only plausible neighborhoods of wind turbines, encoding prior beliefs about which relational dependencies are relevant for the forecasting task. The resulting model is:

$$p_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{i}_{t:t+h} \mid \mathcal{G}_{t-W:t}\right) \tag{4.3}$$

where $\mathcal{G}_t = \langle \boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{M}_t, \mathcal{E}_t, \boldsymbol{V} \rangle$ denotes all the information available to the model at time t (Section 2.2). In this specific application, the relational graph is assumed to be time-invariant and non-attributed (i.e., $\mathcal{E}_t = \mathcal{E}$ for all time steps t).

4.1.3 Relational graph

The SDWPF dataset does not provide an explicit graphical representation of the wind farm under analysis. As a result, the relational graph $\mathcal{E}$ must be constructed based on the available data. This section describes two methodologies for constructing $\mathcal{E}$, relying either on the physical proximity or the functional similarity between wind turbines. The time-invariant relational graph $\mathcal{E}$ is encoded in the form of an adjacency matrix $\boldsymbol{A} \in [0,1]^{N\times N}$, where each entry A_{ij} quantifies the strength of the relational dependency existing between sensors i and j. Formally, $\mathcal{E}$ is defined as:

$$\mathcal{E} = \{(i,j) \mid A_{ij} \geq \tau, \quad i,j = 1,\dots,N\} \tag{4.4}$$

where $\tau \in [0,1]$ is a threshold hyperparameter controlling graph's connectivity retention. The adjacency matrix $\boldsymbol{A}$ encodes prior beliefs about which relational dependencies are relevant for the forecasting task. It follows that the model is exposed to different inductive biases depending on how $\boldsymbol{A}$ is generated, influencing the learning process accordingly.

Proximity relational graph

This method involves generating a relational adjacency matrix relying on the physical proximity between wind turbines (Figure 4.2), encoding information about the mutual influence of neighbouring sensors. Let the position of turbine i be described by the coordinate vector $\boldsymbol{v}^i = (\chi^i, \gamma^i)$, where χ^i and γ^i denote horizontal (`Xpos`) and vertical (`Ypos`) coordinates, respectively. The pairwise Euclidean distance matrix $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ between turbines is defined as:

$$D_{ij} = \|\boldsymbol{v}^i - \boldsymbol{v}^j\|_2 = \sqrt{(\chi^i - \chi^j)^2 + (\gamma^i - \gamma^j)^2} \quad \forall i,j = 1,\dots,N \tag{4.5}$$

$\boldsymbol{D}$ is then converted into a proximity matrix applying a Gaussian kernel to each of its elements. The resulting relational adjacency matrix $\boldsymbol{A} \in [0,1]^{N\times N}$ is defined as:

$$A_{ij} = \exp\left(-\frac{D_{ij}^2}{2\,\sigma_{\boldsymbol{D}}^2}\right) \quad \forall i, j = 1, \ldots, N \tag{4.6}$$

where $\sigma_{\boldsymbol{D}}$ denotes the scaling parameter controlling the width of the kernel, corresponding to the standard deviation of the entries of $\boldsymbol{D}$.

Functional relational graph

This method involves generating a relational adjacency matrix by comparing behavioral and environmental patterns between wind turbines (Figure 4.3), encoding information about sensors functional similarities regardless of their physical location. Let $\hat{\boldsymbol{u}}^i$ be the column vector produced by flattening the covariates matrix $\boldsymbol{U}^i$ over the temporal dimension:

$$\hat{\boldsymbol{u}}^i = [u_{1,1}^i, \ldots, u_{1,d_u}^i, \ldots, u_{T,d_u}^i]^\top \in \mathbb{R}^{T \cdot d_u} \tag{4.7}$$

where $u_{t,k}^i$ denotes the value of the k-th covariate at time t for turbine i. The relational adjacency matrix $\boldsymbol{A} \in [0,1]^{N \times N}$ is defined as:

$$A_{ij} = \cos\left(\hat{\boldsymbol{u}}^i, \hat{\boldsymbol{u}}^j\right) = \frac{{\hat{\boldsymbol{u}}^i}^\top \hat{\boldsymbol{u}}^j}{\|\hat{\boldsymbol{u}}^i\|_2 \, \|\hat{\boldsymbol{u}}^j\|_2} \quad \forall i, j = 1, \ldots, N \tag{4.8}$$

4.2 Training and evaluation

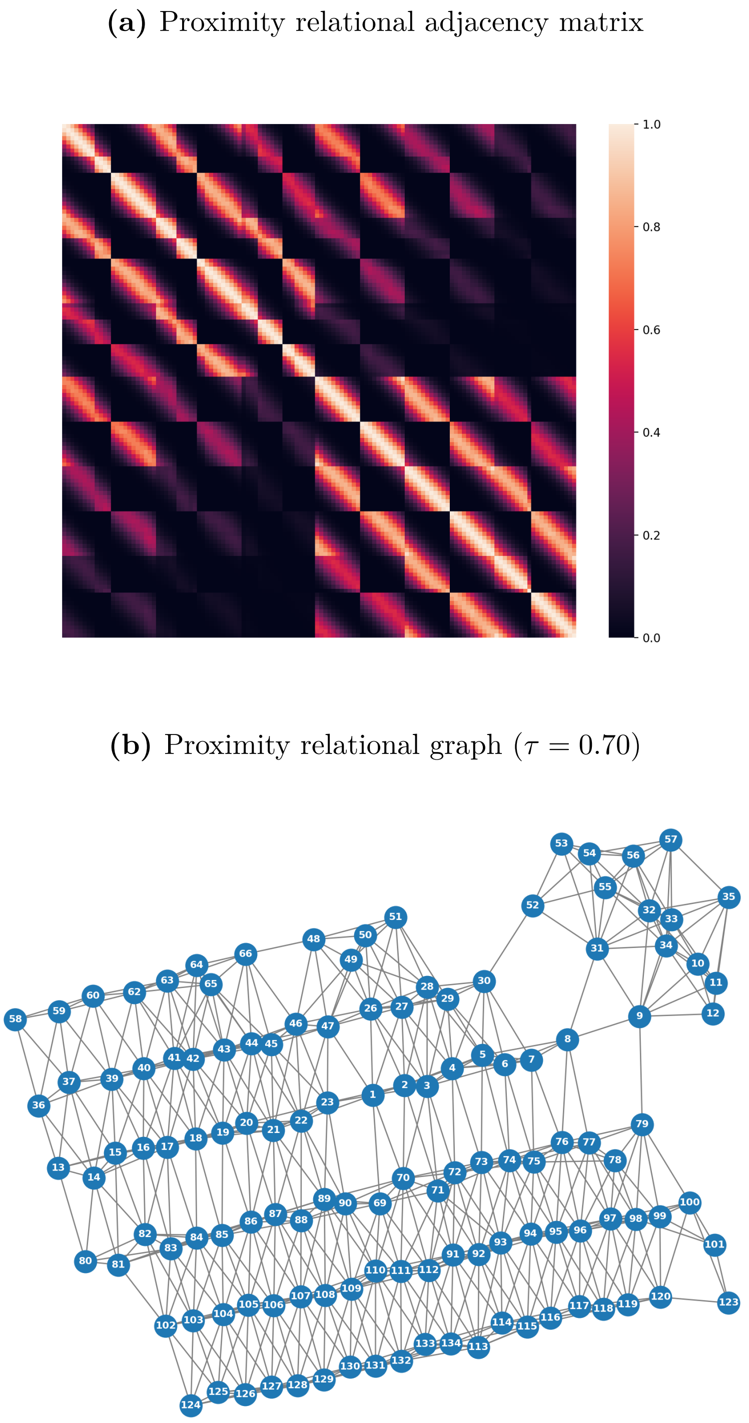
The wind energy production multi-step ahead forecasting problem is addressed as a point estimation task. In this context, a model $\mathcal{F}$ parameterized by learnable parameters $\boldsymbol{\theta}$ generates predictions such that:

$$\hat{\boldsymbol{X}}_{t:t+H} = \mathcal{F}\left(\mathcal{G}_{t-W:t}; \boldsymbol{\theta}\right) \quad s.t. \quad \hat{\boldsymbol{X}}_{t:t+H} \approx \mathbb{E}\left[\boldsymbol{X}_{t:t+H}\right] \tag{4.9}$$

The learnable parameters $\boldsymbol{\theta}$ are estimated by minimizing a generic differentiable loss function $\mathcal{L}(\cdot)$ between the predicted values $\hat{\boldsymbol{X}}_{t:t+H}$ and the true observations $\boldsymbol{X}_{t:t+H}$ over a training dataset:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \frac{1}{T} \sum_{t=1}^{T} \mathcal{L}\left(\boldsymbol{X}_{t:t+H}, \hat{\boldsymbol{X}}_{t:t+H}\right) \tag{4.10}$$

For Spatio-Temporal Graph Neural Network, loss minimization is carried out using backpropagation [22]. This section describes the methodological framework used for the training and evaluation of the models under analysis, covering the adopted evaluation metrics, loss function and optimizer, dataset splitting strategies, and model retention policies.

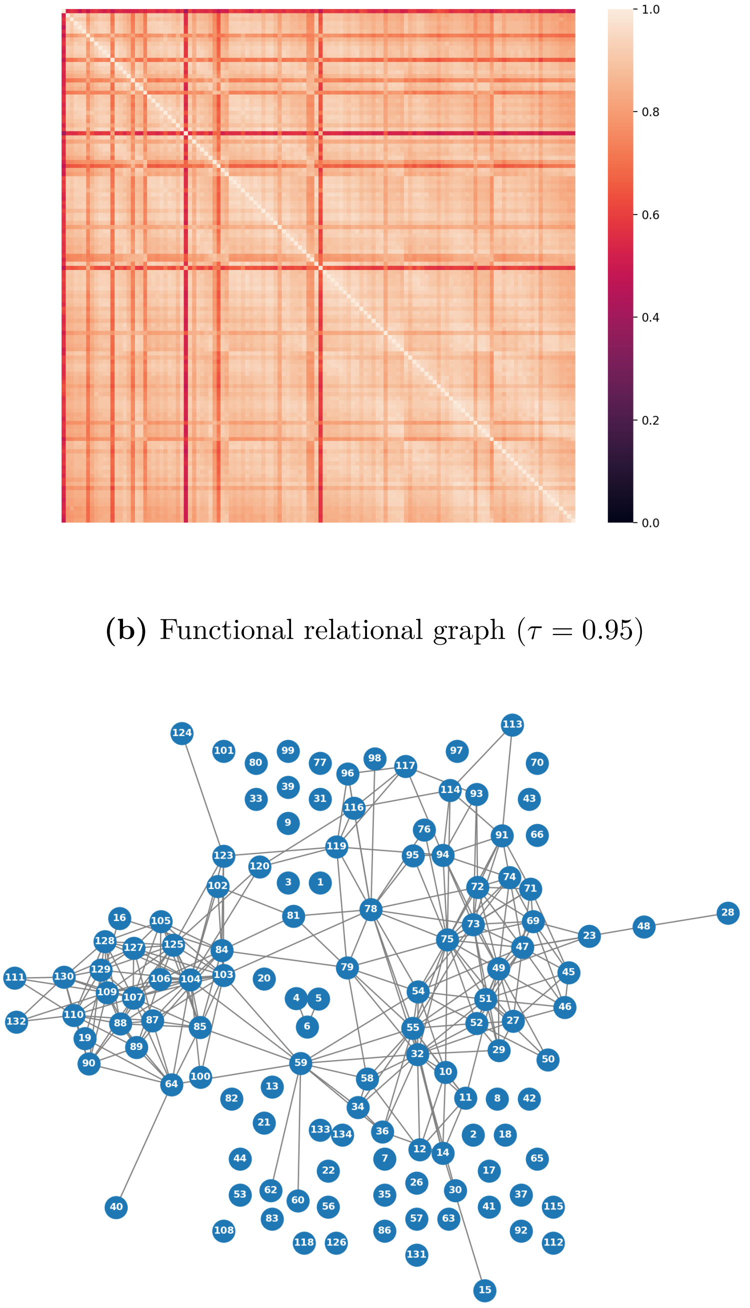


(a) Proximity relational adjacency matrix

(b) Proximity relational graph ($\tau = 0.70$)

Figure 4.2: Proximity relational adjacency matrix and graph

(a) Functional relational adjacency matrix



(b) Functional relational graph ($\tau = 0.95$)

Figure 4.3: Functional relational adjacency matrix and graph

4.2.1 Evaluation metrics

This section introduces a robust implementations of the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) measures, designed to evaluate models' predictive performances handling the presence of missing records by incorporating mask information. These metrics are applied at both local and global scales, assessing predictions for each individual wind turbine and for the wind farm as a whole, respectively.

Masked wind turbine evaluation metrics

Wind turbine MAE and RMSE are defined as follows:

$$\mathrm{MAE}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}, \boldsymbol{M}\right) := \frac{1}{N}\sum_{n=1}^{N} \frac{\sum\limits_{t=1}^{T}\sum\limits_{h=0}^{H-1} m_{t+h}^{n}\left|x_{t+h}^{n} - \hat{x}_{t+h}^{n}\right|}{\sum\limits_{t=1}^{T}\sum\limits_{h=0}^{H-1} m_{t+h}^{n}} \tag{4.11}$$

$$\mathrm{RMSE}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}, \boldsymbol{M}\right) := \frac{1}{N}\sum_{n=1}^{N} \sqrt{\frac{\sum\limits_{t=1}^{T}\sum\limits_{h=0}^{H-1} m_{t+h}^{n}\left(x_{t+h}^{n} - \hat{x}_{t+h}^{n}\right)^{2}}{\sum\limits_{t=1}^{T}\sum\limits_{h=0}^{H-1} m_{t+h}^{n}}} \tag{4.12}$$

In addition to the overall performance, it is possible to analyze the models accuracy focusing on specific prediction horizons in order to understand how prediction errors evolve over different lead times:

$$\mathrm{MAE}_{h}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}, \boldsymbol{M}\right) := \frac{1}{N}\sum_{n=1}^{N} \frac{\sum\limits_{t=1}^{T} m_{t+h}^{n}\left|x_{t+h}^{n} - \hat{x}_{t+h}^{n}\right|}{\sum\limits_{t=1}^{T} m_{t+h}^{n}} \tag{4.13}$$

$$\mathrm{RMSE}_{h}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}, \boldsymbol{M}\right) := \frac{1}{N}\sum_{n=1}^{N} \sqrt{\frac{\sum\limits_{t=1}^{T} m_{t+h}^{n}\left(x_{t+h}^{n} - \hat{x}_{t+h}^{n}\right)^{2}}{\sum\limits_{t=1}^{T} m_{t+h}^{n}}} \tag{4.14}$$

Masked wind farm evaluation metrics

Computing the wind farm metrics requires aggregating data while handling cases where some wind turbines may be affected by outliers. This is achieved by combining the individual wind turbine masks into a single wind farm mask $\bar{m}_t$, ensuring that predictions are evaluated only if at least one turbine shows valid records at time t:

$$\bar{m}_t = \begin{cases} 1 & \text{if } \sum\limits_{n=1}^{N} m_t^n > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.15}$$

The resulting wind farm MAE and RMSE are defined as follows:

$$\overline{\text{MAE}}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}, \boldsymbol{M}\right) := \frac{\sum\limits_{t=1}^{T} \sum\limits_{h=0}^{H-1} \bar{m}_{t+h} \left| \sum\limits_{n=1}^{N} x_{t+h}^n - \sum\limits_{n=1}^{N} \hat{x}_{t+h}^n \right|}{\sum\limits_{t=1}^{T} \sum\limits_{h=0}^{H-1} \bar{m}_{t+h}} \tag{4.16}$$

$$\overline{\text{RMSE}}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}, \boldsymbol{M}\right) := \sqrt{\frac{\sum\limits_{t=1}^{T} \sum\limits_{h=0}^{H-1} \bar{m}_{t+h} \left(\sum\limits_{n=1}^{N} x_{t+h}^n - \sum\limits_{n=1}^{N} \hat{x}_{t+h}^n \right)^2}{\sum\limits_{t=1}^{T} \sum\limits_{h=0}^{H-1} \bar{m}_{t+h}}} \tag{4.17}$$

Again, it is possible to analyze the model's accuracy focusing on specific forecast horizons in order to understand how prediction errors evolve over different lead times:

$$\overline{\text{MAE}}_h\left(\boldsymbol{X}, \hat{\boldsymbol{X}}, \boldsymbol{M}\right) := \frac{\sum\limits_{t=1}^{T} \bar{m}_{t+h} \left| \sum\limits_{n=1}^{N} x_{t+h}^n - \sum\limits_{n=1}^{N} \hat{x}_{t+h}^n \right|}{\sum\limits_{t=1}^{T} \bar{m}_{t+h}} \tag{4.18}$$

$$\overline{\text{RMSE}}_h\left(\boldsymbol{X}, \hat{\boldsymbol{X}}, \boldsymbol{M}\right) := \sqrt{\frac{\sum\limits_{t=1}^{T} \bar{m}_{t+h} \left(\sum\limits_{n=1}^{N} x_{t+h}^n - \sum\limits_{n=1}^{N} \hat{x}_{t+h}^n \right)^2}{\sum\limits_{t=1}^{T} \bar{m}_{t+h}}} \tag{4.19}$$

4.2.2 Loss function

The loss function adopted to learn $\boldsymbol{\theta}$ is the masked Mean Absolute Error at the wind turbine level, as defined in Eq. 4.11:

$$\mathcal{L}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}, \boldsymbol{M}\right) = \text{MAE}\left(\boldsymbol{X}, \hat{\boldsymbol{X}}, \boldsymbol{M}\right) := \frac{1}{N} \sum_{n=1}^{N} \frac{\sum\limits_{t=1}^{T} \sum\limits_{h=0}^{H-1} m_{t+h}^n \left| x_{t+h}^n - \hat{x}_{t+h}^n \right|}{\sum\limits_{t=1}^{T} \sum\limits_{h=0}^{H-1} m_{t+h}^n} \tag{4.20}$$

By incorporating mask information in the loss function, gradients are only calculated for valid data points during backpropagation, preventinting the model from updating its parameters based invalid information.

4.2.3 Optimizer

The optimizer adopted to learn $\boldsymbol{\theta}$ is Adam [23]. Adam is an adaptive optimization algorithm that adjusts the learning rate for each parameter based on estimates of the first and second moments of the gradients. The optimizer updates the model parameters $\boldsymbol{\theta}$ at each time step k with learning rate α:

$$\theta_k = \theta_{k-1} - \alpha \frac{\hat{m}_k}{\sqrt{\hat{v}_k} + \epsilon} \tag{4.21}$$

Where m_k and v_k denote the estimates of the first and second moments of the gradients:

$$m_k = \beta_1 m_{k-1} + (1 - \beta_1) g_k, \quad v_k = \beta_2 v_{k-1} + (1 - \beta_2) g_k^2 \tag{4.22}$$

Scaled by β_1 and β_2 controlling the exponential decay rates of the moment estimates, respectively:

$$\hat{m}_k = \frac{m_k}{1 - \beta_1^k}, \quad \hat{v}_k = \frac{v_k}{1 - \beta_2^k} \tag{4.23}$$

4.2.4 Dataset splitting strategy

To evaluate the performance of the model, the dataset is divided into training and test sets. The adopted splitting strategy preserves the chronological order of the dataset and avoids shuffling, ensuring temporal consistency and preventing potential future information leakage. The dataset is split such that the first 80% of the data is used for training, and the remaining 20% is used for testing purposes.

4.2.5 Model retention policy

The adopted model retention policy relies on a best model checkpointing strategy, where the model state is saved whenever the validation loss reaches a new minimum during the training process. This approach works as an offline early stopping mechanism, preventing overfitting by ensuring that the best-performing model state is checkpointed as long as validation performance improves.

4.3 Experiments

This section presents the adopted experimental framework, designed to investigate the contribution of temporal and spatial information to forecasting accuracy. The experiments focus on three different classes of learners, characterized by incremental degrees of spatiotemporal awareness:

Baseline learners The models included in this class of learners produce a forecast for each individual wind turbine without considering temporal dependencies or spatial relationships among generators. Given a prediction horizon H, point estimation task for each turbine i is formulated as:

$$\hat{\boldsymbol{x}}^i_{t+h} = \mathcal{F}^i_h\left(\boldsymbol{x}^i_{t-W:t}, \boldsymbol{u}^i_{t-W:t}; \boldsymbol{\theta}\right) \quad s.t. \quad \hat{\boldsymbol{x}}^i_{t+h} \approx \mathbb{E}\left[\boldsymbol{x}^i_{t+h}\right] \tag{4.24}$$

Temporal learners The models included in this class of learners produce a global forecast that leverages temporal dependencies remaining agnostic to spatial relationships among wind turbines. Given a prediction horizon H, the point estimation task for the whole wind farm is formulated as:

$$\hat{\boldsymbol{X}}_{t:t+H} = \mathcal{F}\left(\boldsymbol{X}_{t-W:t}, \boldsymbol{U}_{t-W:t}; \boldsymbol{\theta}\right) \quad s.t. \quad \hat{\boldsymbol{X}}_{t:t+H} \approx \mathbb{E}\left[\boldsymbol{X}_{t:t+H}\right] \tag{4.25}$$

Spatiotemporal learners The models included in this class of learners leverage both temporal and spatial information. Given a prediction horizon H, the point estimation task for the whole wind farm is formulated as Eq. 4.26 and reported here for comparison purposes:

$$\hat{\boldsymbol{X}}_{t:t+H} = \mathcal{F}\left(\mathcal{G}_{t-W:t}; \boldsymbol{\theta}\right) \quad s.t. \quad \hat{\boldsymbol{X}}_{t:t+H} \approx \mathbb{E}\left[\boldsymbol{X}_{t:t+H}\right] \tag{4.26}$$

The proposed experiments evaluate different models within each class of learners by comparing their accuracy in the wind energy production multi-step ahead forecasting task. Each of these model, summarized in Table 4.1, is evaluated on multiple prediction tasks characterized by different configurations of W and H, testing their performance in both short-term and long-term forecasting exercises.

Model	Temporal	Spatial	Label
Multi-target Linear Regression	-	-	MLR
Multi-target Random Forest	-	-	MRF
Recurrent Neural Network	●	-	RNN
Transformer	●	-	TSF
Gated Graph Network	●	●	GGN
Graph WaveNet	●	●	GWN

Table 4.1: Models summary

4.3.1 Linear regression

Model overview The Linear Regression (LR) is a statistical method used to model the relationship between a dependent variable and one or more independent variables by fitting a linear model to the observed data. In the context of time series forecasting, it can be applied to predict future values by learning the linear relationships between historical observations and future outputs.

Model generalization To support the multi-step ahead forecasting task, the standard formulation is extended to the multi-target setting, which enables the model to predict multiple future time steps simultaneously. Each time step in the prediction horizon is treated as a separate target, and the model is trained to produce forecasts for all targets using the same set of input features.

Dimensionality reduction To address the high dimensionality associated with the increasing size of the backward-looking window W the input data is compressed using Principal Component Analysis. The PCA transforms high-dimensional input into a lower-dimensional space retaining the most important components that capture most of the variability in the data, preserving the information relevant for the prediction task.

Table 4.2 summarizes the Linear Regression configuration used to carry out the experiments described in this section.

Window (W)	Horizon (H)	# of PC
1	1	10
36	1	32
144	1	32
1	36	10
36	36	32
144	36	32
1	144	10
36	144	32
144	144	32

Table 4.2: Linear Regression training configuration

4.3.2 Random forest

Model overview A Random Forest (RF) is an ensemble learning method aggregates the output of multiple decision trees to perform a single estimation. Each tree is trained on a random subset of the data, which helps reduce variance and overfitting, making Random Forest more robust and accurate compared to individual decision trees. A decision tree is constructed by recursively splitting the data based on feature values to create branches, with the goal of minimizing the error in predicting the target variable. At each node of the tree, the model selects the feature and corresponding threshold that best partitions the data into subsets, according to a given criterion (such as MSE for regression tasks). This process continues until a stopping criterion is met, such as a maximum tree depth or a minimum number of samples in a node. Each leaf node in the tree represents a final prediction, that is the average of the target values associated to that node. The recursive partitioning creates a series of non-linear decision boundaries. As a result, decision trees can capture non-linear interactions between features that may not be linearly separable.

Model generalization & dimensionality reduction Following the methodology described in Section 4.3.1, the Random Forest model is extended to handle multi-step ahead forecasting using a multi-target approach. Each time step in the prediction horizon is treated as a separate target, allowing the model to predict multiple future time steps simultaneously. Similarly, PCA is used to handle the high dimensionality associated with large backward-looking windows.

Table 4.3 summarizes the Random Forest configuration used to carry out the experiments described in this section.

Window (W)	Horizon (H)	# of estimators	# of PC
1	1	5	10
36	1	5	32
144	1	5	32
1	36	5	10
36	36	5	32
144	36	5	32
1	144	5	10
36	144	5	32
144	144	5	32

Table 4.3: Random Forest training configuration

4.3.3 Recurrent Neural Network

Model overview A Recurrent Neural Network is a deep learning architecture designed to process sequential data by maintaining a hidden state which is updated at each time step based on the current input and previous hidden representation. This structure allows the network to learn from past inputs and represent temporal dependencies within sequences of data. A traditional RNN architecture consists of stacked recurrent layers, followed by a fully connected read-out layer to generate multi-step ahead predictions.

Information gating mechanism LSTM [24] is an extension of the vanilla RNN cell that relies on a set of gating mechanisms to manage the information flow in the network. These gates allow the model to select the information to retain or discard, mitigating vanishing or exploding gradient issues that might arise when learning long input temporal sequences. Specifically, the architecture of the LSTM is built around three main gates: the forget date, deciding what information from the previous cell state to discard; the input gate determining which new information to add to the cell state; and the output gate controlling the output based on the current cell state.

Table 4.4 summarizes the Recurrent Neural Network configuration used to carry out the experiments described in this section.

Window (W)	Horizon (H)	Batch size	Epochs	Optimizer	Learning rate
1	1	512	30	Adam	0.000500
36	1	128	30	Adam	0.000125
144	1	64	30	Adam	0.000075
1	36	512	30	Adam	0.000500
36	36	128	30	Adam	0.000125
144	36	32	30	Adam	0.000040
1	144	512	30	Adam	0.000500
36	144	64	30	Adam	0.000075
144	144	32	30	Adam	0.000040

Table 4.4: Recurrent Neural Network training configuration

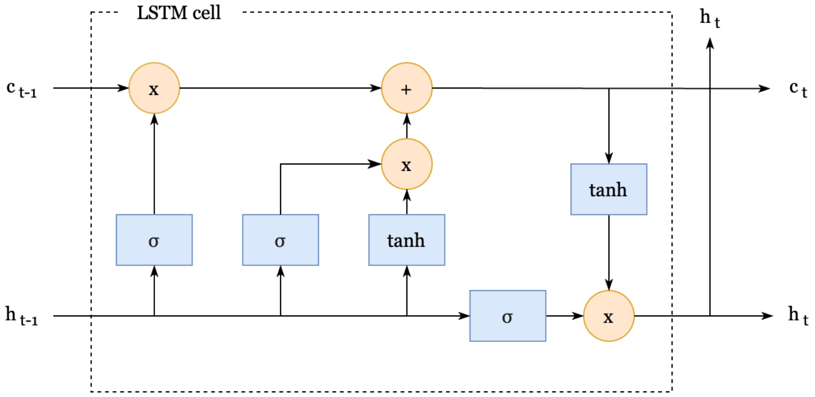


Figure 4.4: LSTM cell

4.3.4 Transformer

Model overview The Transformer is a deep learning architecture designed for representing relational dependencies in sequential data. Unlike Recurrent Neural Networks, which process inputs in a sequential order, the Transformer relies on a self-attention mechanism that allows the model to consider the entire input sequence at once. This makes the model able to capture both short-term and long-term relational patterns and a suitable alternative for time series forecasting tasks.

Multi-head self-attention mechanism The principal component of the Transformer is the multi-head self-attention mechanism, which assigns importance scores between all pairs of elements in the input sequence based on their relevance to the prediction. Being agnostic of the sequential order of input signals, the Transformer relies on a positional encoding layer that adds information about the relative positions of elements in the input sequence, informing the model about the temporal continuity.

Table 4.5 summarizes the Transformer configuration used to carry out the experiments described in this section.

Window (W)	Horizon (W)	Batch size	Epochs	Optimizer	Learning rate
1	1	512	10	Adam	0.000500
36	1	64	10	Adam	0.000050
144	1	16	10	Adam	0.000010
1	36	512	10	Adam	0.000500
36	36	64	10	Adam	0.000050
144	36	16	10	Adam	0.000010
1	144	512	10	Adam	0.000500
36	144	64	10	Adam	0.000050
144	144	16	10	Adam	0.000010

Table 4.5: Transformer training configuration

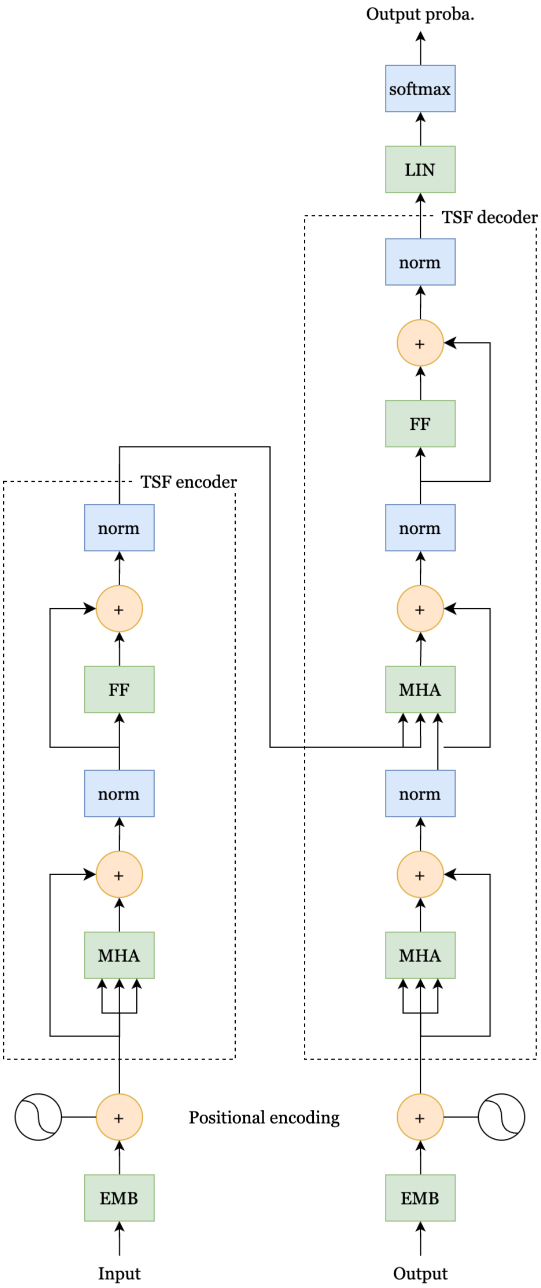


Figure 4.5: Transformer cell

4.3.5 Gated Graph Network

Model overview The Gated Graph Network is a neural network inspired by the FC-GNN [25] architecture introduced by Satorras et al. to model spatiotemporal dependencies in multivariate time series data. This network belongs to the Time-then-Space STGNNs taxonomy described earlier in Section 2.3.

Encoder & decoder architecture The architecture consists of an encoder and a decoder. The encoder processes each input time series independently, producing an embedding for each time series. These embeddings are then passed through a GNN consisting of L graph convolutional layers [26] that incorporates an attention mechanism that dynamically assigns weights to the graph edges, representing the strength of relational dependency across nodes. The node embeddings produced by the GNN are then fed into the decoder, which reconstructs the predictions for the target prediction horizon.

Table 4.6 summarizes the Gated Graph Network configuration used to carry out the experiments described in this section.

Window (W)	Horizon (H)	Batch size	Epochs	Optimizer	Learning rate
1	1	1024	20	Adam	0.000500
36	1	1024	20	Adam	0.000250
144	1	512	20	Adam	0.000100
1	36	1024	20	Adam	0.000500
36	36	1024	20	Adam	0.000250
144	36	512	20	Adam	0.000100
1	144	1024	20	Adam	0.000500
36	144	1024	20	Adam	0.000250
144	144	512	20	Adam	0.000100

Table 4.6: Gated Graph Network training configuration

4.3.6 Graph WaveNet

Model overview The Graph WaveNet [27] is a deep learning architecture designed to model long-term spatiotemporal dependencies in graph-structured data. Inspired by the WaveNet [28] architecture, Graph WaveNet relies on dilated causal convolutional layers to capture long-term relationships. Each layer is composed of a graph convolutional layer for spatial dependencies and a gated temporal convolutional layer to model temporal relationships. The receptive field of these convolutions grows exponentially as the number of layers increases enabling the model to handle long sequences without occurring in vanishing or exploding gradients and decreasing computational requirements.

Table 4.7 summarizes the Graph WaveNet configuration used to carry out the experiments described in this section.

Window (W)	Horizon (H)	Batch size	Epochs	Optimizer	Learning rate
1	1	256	20	Adam	0.000250
36	1	64	20	Adam	0.000075
144	1	32	20	Adam	0.000015
1	36	256	20	Adam	0.000250
36	36	64	20	Adam	0.000075
144	36	32	20	Adam	0.000015
1	144	256	20	Adam	0.000250
36	144	64	20	Adam	0.000075
144	144	32	20	Adam	0.000015

Table 4.7: Graph WaveNet training configuration

Self-adaptive adjacency matrix

Traditional graph-based models typically rely on a fixed graphical representation, where node relationships, encoded in the form of an adjacency matrix, are predefined and static throughout the learning process. While effective in cases where node relationships are well known, such a static graph can limit the model's ability to have a representation of dynamic or latent relational dependencies that could improve prediction accuracy. Graph WaveNet addresses this limitation relying on a self-adaptive adjacency matrix built using trainable node embeddings, which are updated via backpropagation as part of the end-to-end learning process.

Figure 4.7 shows an example of a learned self-adaptive adjacency matrix and the resulting graph structure in the context of the wind energy production forecasting exercise. By applying the PageRank [29] algorithm to the learned adjacency matrix it is possible to identify the most influential turbines in the wind farm, corresponding to $\texttt{Turb}_3$ and $\texttt{Turb}_{131}$, respectively representing prototype wind turbines for left and right-hand sides partitions of the wind farm (Figure 4.8). While the exact geographical locations of these wind turbines are not known, and it is difficult to definitively attribute physical meanings to these prototypes, the learned adjacency matrix suggests the presence of important functional relationships between these turbines. Further analysis of these findings is beyond the scope of this study.

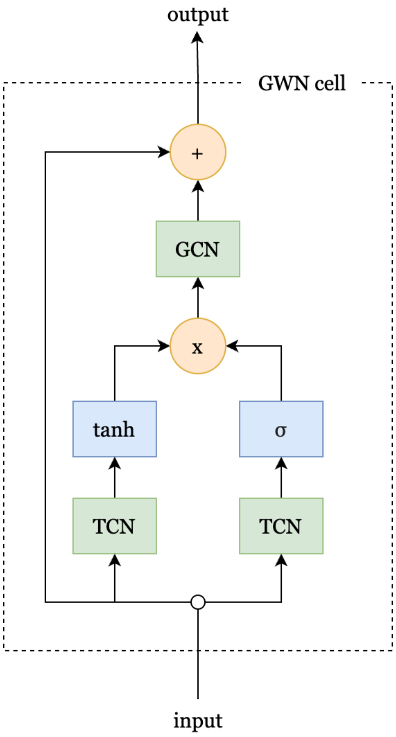
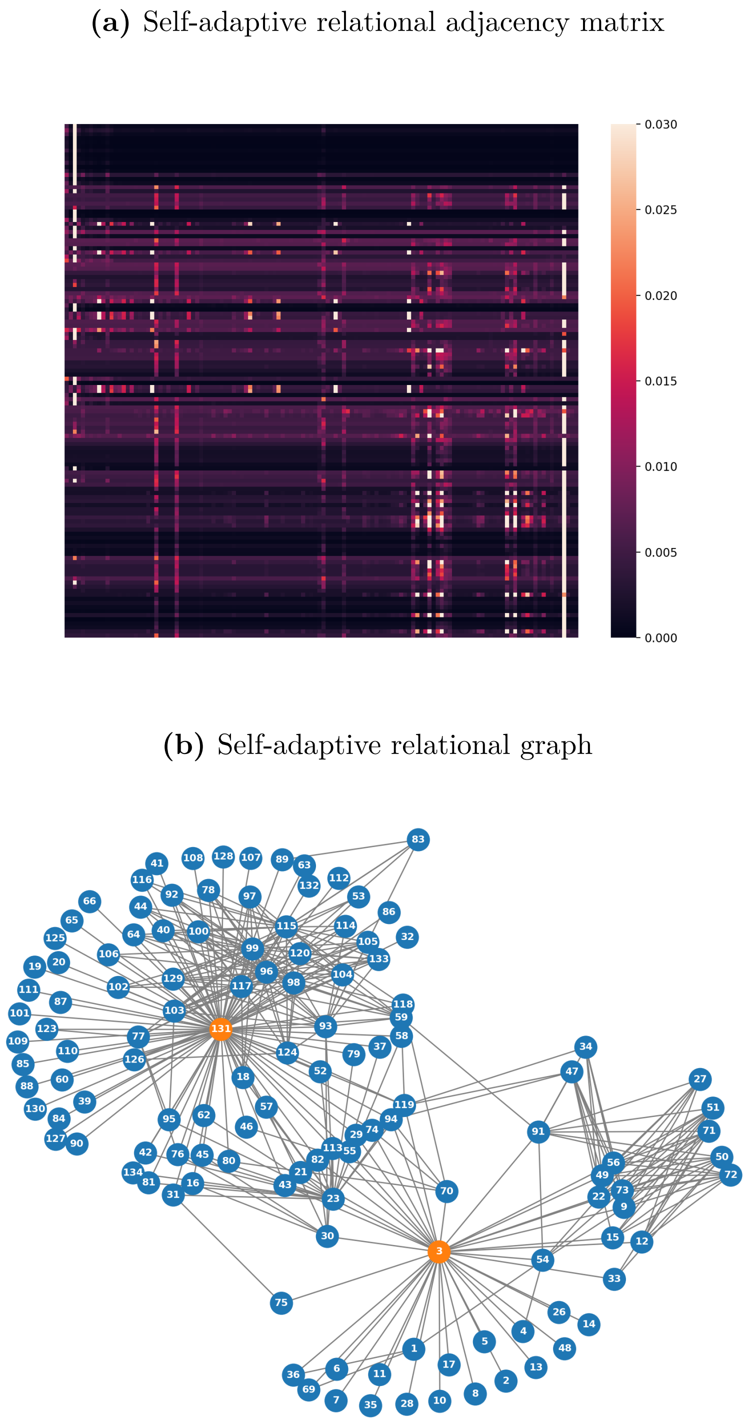


Figure 4.6: Graph WaveNet cell



(a) Self-adaptive relational adjacency matrix

(b) Self-adaptive relational graph

Figure 4.7: Self-adaptive relational adjacency matrix and graph

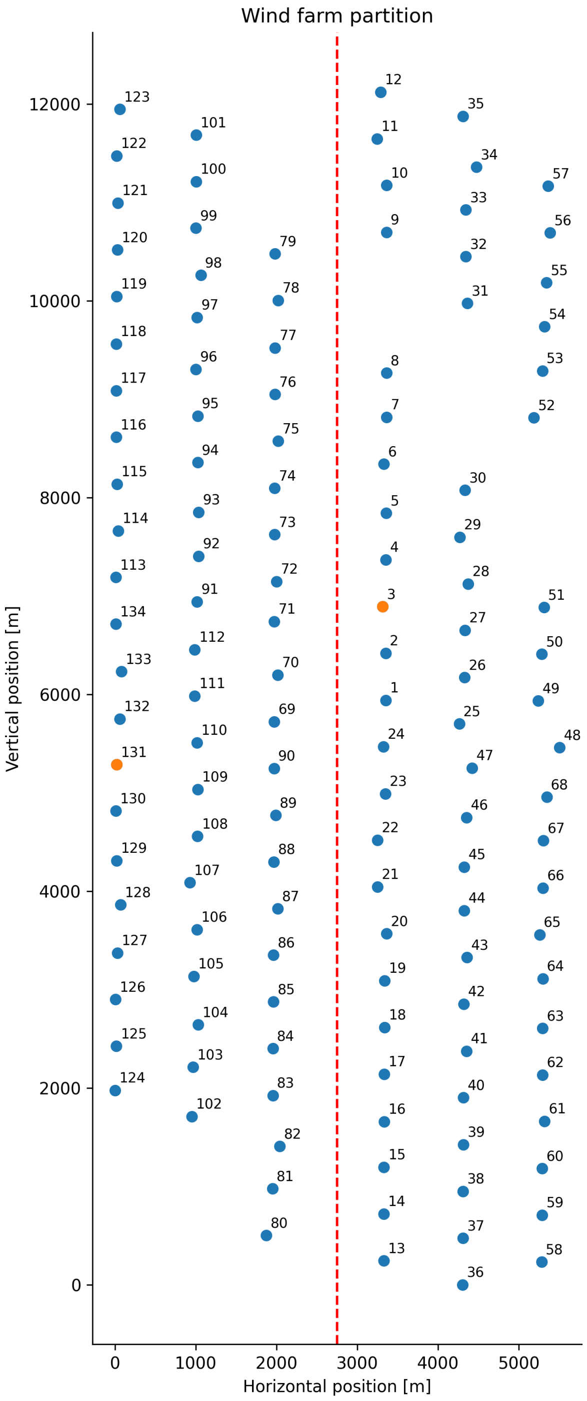


Figure 4.8: Wind farm partition

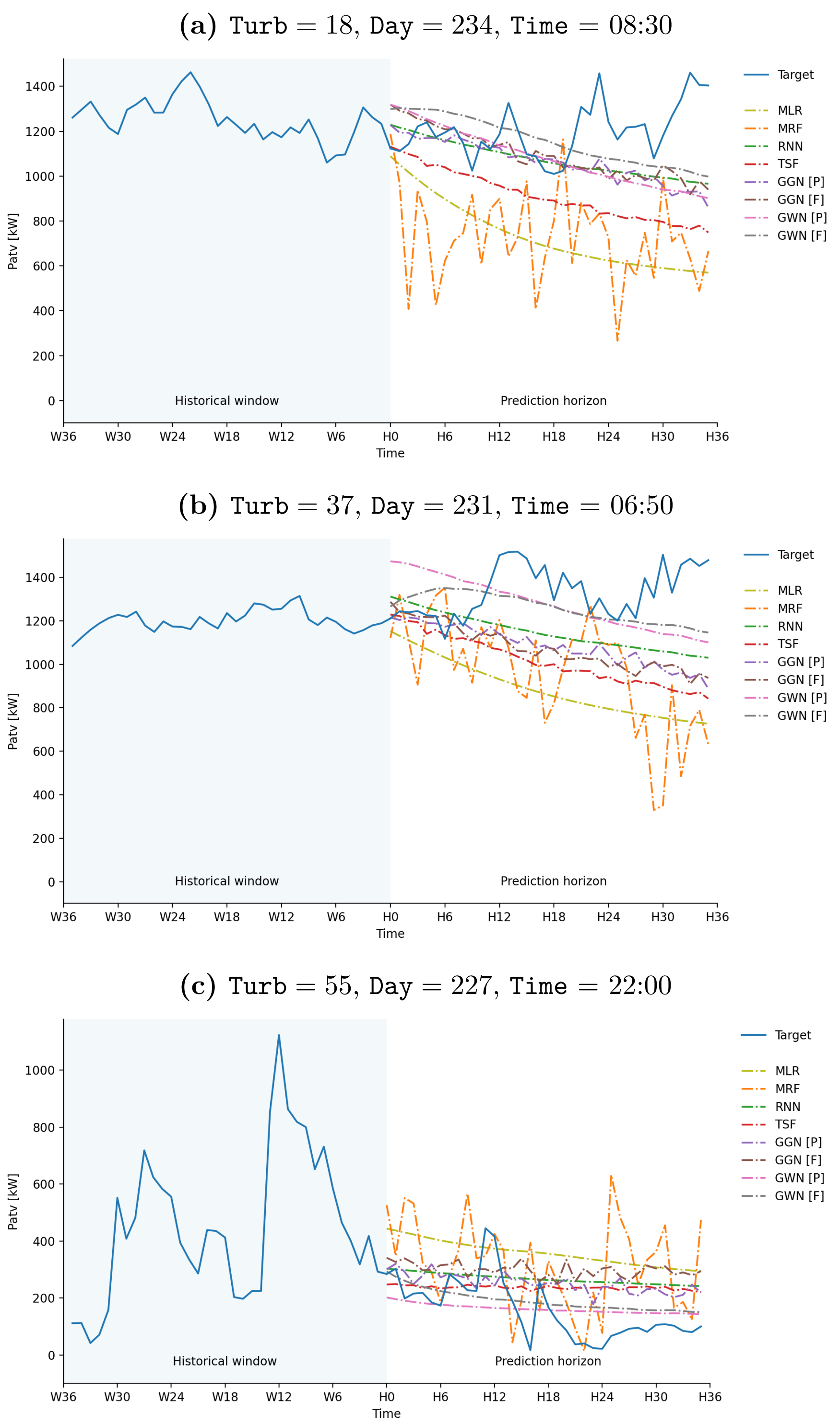
4.4 Comparative analysis

This section presents a comparative analysis of the experiments introduced in Section 4.3. The experiments aim at providing an evaluation of the effectiveness of STGNNs on short and long term wind energy production forecasting tasks while also investigating the influence of different configurations of relational information on spatiotemporal models. The models listed in Table 4.1 were trained and tested across nine different scenarios, with historical window sizes W and forecasting horizons H set to 1, 36, and 144, corresponding to ten-minute, six-hour, and one-day ahead forecasting tasks, respectively. Additionally, the spatiotemporal architectures were provided with relational information encoded as either proximity (Section 4.1.3, denoted as `P`) or functional (Section 4.1.3, denoted as `F`) relational graphs. The models' predictive performance was tested at both local and global level, relying on the masked wind turbine (Section 4.2.1) and the masked wind farm (Section 4.2.1) evaluation metrics, respectively. The results for individual wind turbines are outlined in Tables 4.8, 4.9, and 4.10, while the wind farm-level evaluations are summarized in Tables 4.11, 4.12, and 4.13 for each prediction horizon, respectively. Each table also reports validation metrics focusing on individual prediction steps in order to evaluate model performances on short ($h = 1$), middle ($h = H/2$) and long ($h = H$) prediction horizons.

Predictive performance From Tables 4.8 and 4.11 it is possible to observe that all the models show similar performances in short-term prediction tasks. However, as the prediction horizon increases, temporal and spatiotemporal learners show better forecast accuracy, as summarized in Tables 4.9, 4.12, 4.10, and 4.13. Specifically, STGNNs tend to be the best predictors among the competitors, with Graph WaveNet consistently outperforming the other models on both short and long-term prediction tasks.

Prediction horizon The predictive performance of all models generally generally degrades as the forecasting horizon increases, suggesting that the energy output profile cannot be fully described in terms of available historical data. Prediction accuracy mostly deteriorates on prediction horizons larger than six hours, according to the findings of the autocorrelation analysis presented in Section 3.3.4. Figures 4.9 and 4.10 illustrate this behavior, reporting some forecast samples extracted from the dataset.

Relational information The STGNNs configured to consume the functional relational information tend to show better predictive capabilities than their counterpart, configured, instead, to consume proximity relational information. This suggests that the functional relationships are better suited in retaining relevant relational dependencies for the wind energy prediction task.

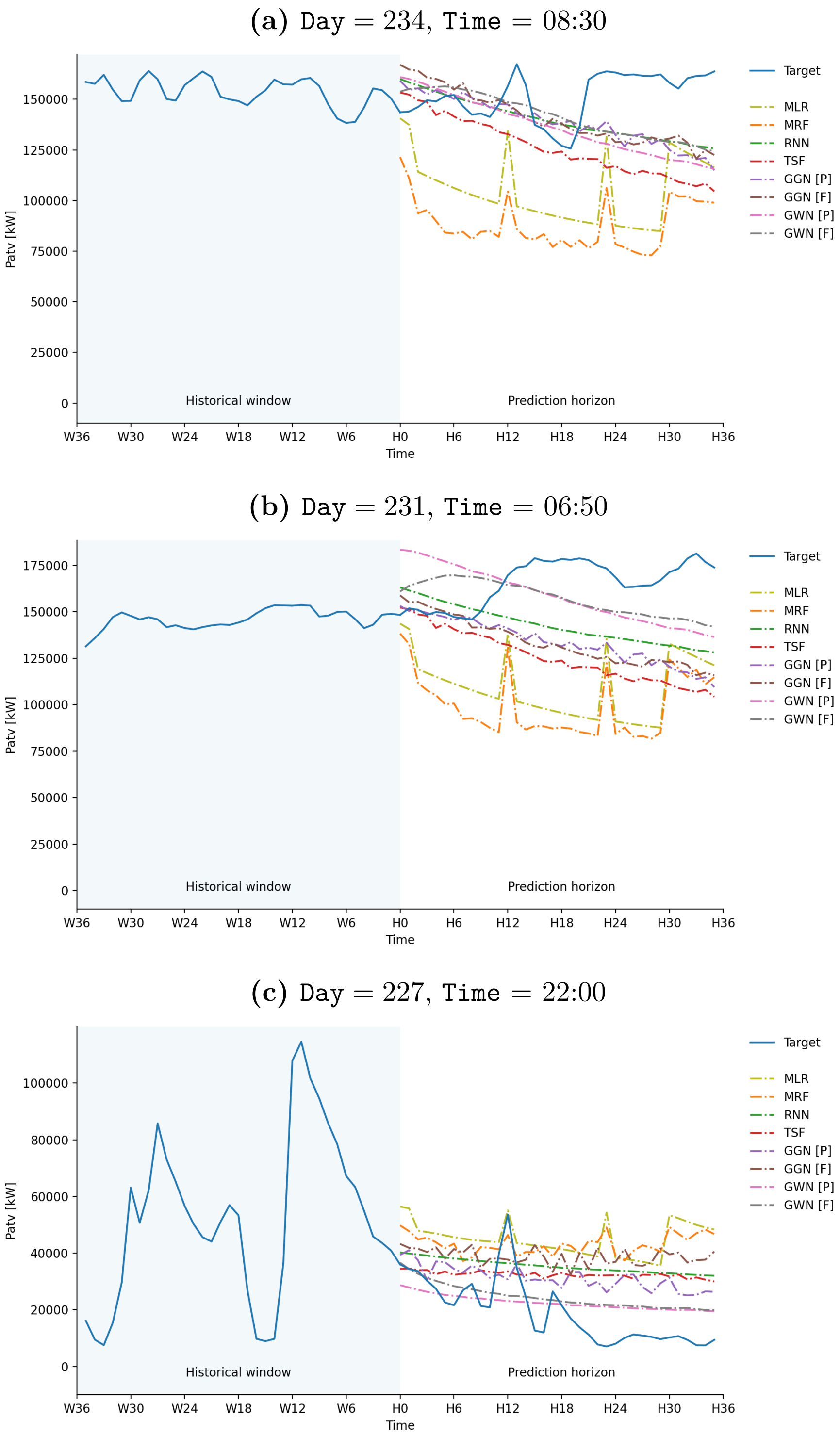


(a) Turb = 18, Day = 234, Time = 08:30

(b) Turb = 37, Day = 231, Time = 06:50

(c) Turb = 55, Day = 227, Time = 22:00

Figure 4.9: Wind turbines power output predictions samples



(a) `Day` = 234, `Time` = 08:30

(b) `Day` = 231, `Time` = 06:50

(c) `Day` = 227, `Time` = 22:00

Figure 4.10: Wind farm power output predictions samples

	Model	Graph	Mean Absolute Error [kW]				Root Mean Squared Error [kW]			
			Avg.	$h = 1$	$h = H/2$	$h = H$	Avg.	$h = 1$	$h = H/2$	$h = H$
$W = 1$	MLR	-	59.24	59.24	59.24	59.24	92.03	92.03	92.03	92.03
	MFR	-	114.89	114.89	114.89	114.89	166.43	166.43	166.43	166.43
	RNN	-	65.30	65.30	65.30	65.30	101.54	101.54	101.54	101.54
	TSF	-	55.41	55.41	55.41	55.41	91.68	91.68	91.68	91.68
	GGN	P	52.44	52.44	52.44	52.44	89.34	89.34	89.34	89.34
	GWN	P	55.76	55.76	55.76	55.76	91.08	91.08	91.08	91.08
	GGN	F	52.37	52.37	52.37	52.37	88.08	88.08	88.08	88.08
	GWN	F	55.60	55.60	55.60	55.60	90.71	90.71	90.71	90.71
$W = 36$	MLR	-	101.46	101.46	101.46	101.46	141.73	141.73	141.73	141.73
	MFR	-	186.10	186.10	186.10	186.10	249.01	249.01	249.01	249.01
	RNN	-	58.37	58.37	58.37	58.37	97.19	97.19	97.19	97.19
	TSF	-	54.61	54.61	54.61	54.61	91.52	91.52	91.52	91.52
	GGN	P	52.04	52.04	52.04	52.04	86.21	86.21	86.21	86.21
	GWN	P	52.49	52.49	52.49	52.49	84.99	84.99	84.99	84.99
	GGN	F	52.01	52.01	52.01	52.01	85.98	85.98	85.98	85.98
	GWN	F	48.71	48.71	48.71	48.71	81.07	81.07	81.07	81.07
$W = 144$	MLR	-	170.06	170.06	170.06	170.06	220.11	220.11	220.11	220.11
	MFR	-	264.66	264.66	264.66	264.66	343.98	343.98	343.98	343.98
	RNN	-	59.73	59.73	59.73	59.73	101.69	101.69	101.69	101.69
	TSF	-	56.45	56.45	56.45	56.45	92.29	92.29	92.29	92.29
	GGN	P	54.26	54.26	54.26	54.26	88.60	88.60	88.60	88.60
	GWN	P	56.85	56.85	56.85	56.85	90.61	90.61	90.61	90.61
	GGN	F	54.95	54.95	54.95	54.95	89.44	89.44	89.44	89.44
	GWN	F	52.39	52.39	52.39	52.39	86.85	86.85	86.85	86.85

Table 4.8: Wind turbines $H = 1$ energy output prediction errors comparison

	Model	Graph	Mean Absolute Error [kW]				Root Mean Squared Error [kW]			
			Avg.	$h = 1$	$h = H/2$	$h = H$	Avg.	$h = 1$	$h = H/2$	$h = H$
$W = 1$	MLR	-	200.14	59.34	240.75	168.88	274.87	92.21	330.58	229.01
	MFR	-	282.76	114.99	322.44	254.78	369.73	167.06	414.83	335.42
	RNN	-	196.92	94.30	207.64	255.91	280.18	146.05	287.86	351.25
	TSF	-	188.49	68.29	200.49	254.30	269.86	102.00	277.42	346.82
	GGN	P	185.82	56.87	199.07	256.52	277.10	91.78	285.57	363.76
	GWN	P	185.75	79.62	196.20	250.78	267.83	117.31	274.27	347.53
	GGN	F	184.60	54.55	198.08	253.47	274.27	89.62	282.24	357.73
	GWN	F	183.55	85.84	191.62	246.75	262.79	131.13	267.26	338.60
$W = 36$	MLR	-	210.51	101.67	247.56	179.18	290.79	142.00	342.51	246.50
	MFR	-	301.66	186.62	343.41	262.16	388.91	249.04	439.02	341.79
	RNN	-	198.63	125.12	205.63	250.32	277.14	172.23	282.09	345.45
	TSF	-	187.65	66.75	199.06	254.51	269.03	101.58	275.33	348.19
	GGN	P	182.55	70.21	193.44	253.47	269.44	104.13	275.44	359.87
	GWN	P	178.11	83.80	186.92	239.38	259.62	128.06	264.96	333.75
	GGN	F	182.78	82.79	191.92	248.13	268.64	118.19	273.68	352.18
	GWN	F	172.43	64.62	183.68	235.06	253.58	95.25	261.01	329.17
$W = 144$	MLR	-	243.54	169.04	271.42	217.68	327.52	219.00	368.24	289.89
	MFR	-	310.76	265.82	326.05	295.24	407.70	345.93	428.55	387.44
	RNN	-	210.72	141.84	216.62	259.35	295.40	196.45	300.28	361.14
	TSF	-	188.54	68.36	199.97	250.64	272.68	102.88	281.66	350.93
	GGN	P	194.73	81.80	205.79	265.98	285.51	114.36	291.60	382.03
	GWN	P	186.84	81.97	197.95	250.80	264.91	117.94	271.91	341.77
	GGN	F	198.33	86.08	212.91	266.66	291.96	124.94	303.39	378.39
	GWN	F	183.00	74.45	194.17	248.16	264.66	114.79	271.10	342.74

Table 4.9: Wind turbines $H = 36$ energy output prediction errors comparison

	Model	Graph	Mean Absolute Error [kW]				Root Mean Squared Error [kW]			
			Avg.	$h = 1$	$h = H/2$	$h = H$	Avg.	$h = 1$	$h = H/2$	$h = H$
$W = 1$	MLR	-	285.92	59.57	265.73	322.08	388.14	92.55	365.83	429.00
	MFR	-	366.05	116.91	362.72	376.74	464.76	169.33	456.12	479.21
	RNN	-	293.75	156.19	303.27	372.48	407.24	207.25	415.61	505.52
	TSF	-	283.09	63.76	304.76	352.94	401.77	101.76	420.96	484.44
	GGN	P	286.71	60.91	305.61	360.71	405.65	92.40	422.24	499.45
	GWN	P	283.98	102.66	303.66	357.72	401.88	146.56	420.44	490.67
	GGN	F	290.15	57.07	310.18	371.62	424.77	92.25	443.53	521.68
	GWN	F	283.09	94.85	302.34	360.92	395.30	139.97	408.23	492.64
$W = 36$	MLR	-	291.08	101.91	267.95	328.65	397.15	142.33	368.91	443.46
	MFR	-	359.51	188.36	371.30	377.72	464.84	252.54	468.98	490.61
	RNN	-	305.48	193.58	309.82	383.96	418.53	246.61	422.12	517.12
	TSF	-	291.92	125.72	308.81	364.60	408.77	159.23	424.85	498.91
	GGN	P	288.10	82.55	312.25	359.77	412.37	114.45	436.52	502.02
	GWN	P	285.39	115.86	304.01	359.52	401.10	164.43	419.59	489.35
	GGN	F	284.23	73.56	308.31	349.76	409.85	107.76	437.48	495.45
	GWN	F	283.23	102.38	303.74	357.92	398.86	150.19	416.85	490.03
$W = 144$	MLR	-	307.44	167.20	287.21	336.51	420.18	216.17	390.65	462.54
	MFR	-	356.51	264.17	335.63	381.41	469.48	344.64	441.20	500.71
	RNN	-	315.25	234.33	316.83	389.02	422.19	298.19	421.69	515.77
	TSF	-	286.50	79.62	308.82	357.54	407.93	117.76	428.20	495.30
	GGN	P	293.89	98.85	313.73	363.75	419.20	133.97	435.53	505.57
	GWN	P	283.71	109.37	299.64	362.27	404.80	149.34	419.34	500.42
	GGN	F	292.79	99.46	312.95	361.01	421.25	139.12	440.14	504.49
	GWN	F	283.09	96.67	302.87	358.72	403.73	139.23	421.57	495.03

Table 4.10: Wind turbines $H = 144$ energy output prediction errors comparison

	Model	Graph	Mean Absolute Error [kW]				Root Mean Squared Error [kW]			
			Avg.	$h = 1$	$h = H/2$	$h = H$	Avg.	$h = 1$	$h = H/2$	$h = H$
$W = 1$	MLR	-	3333.21	3333.21	3333.21	3333.21	5447.06	5447.06	5447.06	5447.06
	MFR	-	5405.79	5405.79	5405.79	5405.79	8557.33	8557.33	8557.33	8557.33
	RNN	-	3562.89	3562.89	3562.89	3562.89	5718.15	5718.15	5718.15	5718.15
	TSF	-	3003.21	3003.21	3003.21	3003.21	5318.08	5318.08	5318.08	5318.08
	GGN	P	2744.69	2744.69	2744.69	2744.69	5140.45	5140.45	5140.45	5140.45
	GWN	P	2966.75	2966.75	2966.75	2966.75	5328.27	5328.27	5328.27	5328.27
	GGN	F	2735.67	2735.67	2735.67	2735.67	5101.76	5101.76	5101.76	5101.76
	GWN	F	2928.30	2928.30	2928.30	2928.30	4834.10	4834.10	4834.10	4834.10
$W = 36$	MLR	-	7054.81	7054.81	7054.81	7054.81	11029.20	11029.20	11029.20	11029.20
	MFR	-	11960.98	11960.98	11960.98	11960.98	16431.39	16431.39	16431.39	16431.39
	RNN	-	3068.69	3068.69	3068.69	3068.69	5253.86	5253.86	5253.86	5253.86
	TSF	-	2906.27	2906.27	2906.27	2906.27	5252.63	5252.63	5252.63	5252.63
	GGN	P	2691.73	2691.73	2691.73	2691.73	4985.41	4985.41	4985.41	4985.41
	GWN	P	2501.63	2501.63	2501.63	2501.63	4517.09	4517.09	4517.09	4517.09
	GGN	F	2715.80	2715.80	2715.80	2715.80	4995.69	4995.69	4995.69	4995.69
	GWN	F	2273.96	2273.96	2273.96	2273.96	3899.88	3899.88	3899.88	3899.88
$W = 144$	MLR	-	13340.61	13340.61	13340.61	13340.61	19632.45	19632.45	19632.45	19632.45
	MFR	-	19781.30	19781.30	19781.30	19781.30	27024.84	27024.84	27024.84	27024.84
	RNN	-	3153.58	3153.58	3153.58	3153.58	5398.98	5398.98	5398.98	5398.98
	TSF	-	3108.15	3108.15	3108.15	3108.15	5423.24	5423.24	5423.24	5423.24
	GGN	P	2885.82	2885.82	2885.82	2885.82	5315.51	5315.51	5315.51	5315.51
	GWN	P	3001.05	3001.05	3001.05	3001.05	5288.21	5288.21	5288.21	5288.21
	GGN	F	3020.82	3020.82	3020.82	3020.82	5470.25	5470.25	5470.25	5470.25
	GWN	F	2570.91	2570.91	2570.91	2570.91	4818.28	4818.28	4818.28	4818.28

Table 4.11: Wind farm $H = 1$ energy output prediction errors comparison

	Model	Graph	Mean Absolute Error [kW]				Root Mean Squared Error [kW]			
			Avg.	$h = 1$	$h = H/2$	$h = H$	Avg.	$h = 1$	$h = H/2$	$h = H$
$W = 1$	MLR	-	16453.18	3335.44	20364.30	13401.33	25591.23	5451.27	31972.13	20279.17
	MFR	-	21201.30	5465.54	25111.73	18358.35	29232.71	8560.84	34389.66	25499.79
	RNN	-	15810.78	5877.51	16990.51	21960.84	24551.04	9400.11	26320.30	33827.21
	TSF	-	15546.11	4305.32	16710.94	22094.54	24070.47	6826.49	25886.84	33692.72
	GGN	P	15293.67	3069.36	16616.79	22392.89	25002.89	5585.16	27106.80	36012.34
	GWN	P	14823.46	5227.18	15724.20	21469.85	23205.55	8238.07	24688.99	33142.67
	GGN	F	15041.79	2841.81	16177.04	21818.81	24000.14	5170.22	25856.37	34281.43
	GWN	F	14911.19	5631.11	15801.44	21420.52	23346.82	9322.40	24839.82	32802.08
$W = 36$	MLR	-	17571.22	7065.01	21151.42	14555.60	27779.17	11037.77	33647.20	22794.17
	MFR	-	22992.93	11975.31	26883.26	19268.48	30724.85	16434.06	35345.74	26475.69
	RNN	-	15975.56	9060.95	16648.05	21133.71	24534.25	13121.08	25687.74	33091.54
	TSF	-	15443.16	3606.00	16642.11	22174.16	24168.03	6357.44	25881.56	34176.09
	GGN	P	14993.59	4545.71	16002.70	22152.71	24366.96	7346.78	26023.46	35672.59
	GWN	P	14360.33	5636.22	15129.31	20401.07	23294.24	9656.72	24667.68	32328.25
	GGN	F	14967.65	5918.21	15699.00	21555.82	24247.40	9058.16	25610.69	34672.50
	GWN	F	13729.27	3605.30	14811.75	19920.29	22370.79	5637.76	24224.89	31667.86
$W = 144$	MLR	-	20624.41	13237.46	23425.70	18006.80	31764.02	19499.30	36374.06	27520.16
	MFR	-	24081.80	19903.19	25522.46	22606.03	34044.24	27158.31	36328.65	31820.63
	RNN	-	17033.96	10803.92	17537.75	21825.10	26578.31	15773.46	27631.98	34675.35
	TSF	-	15581.24	4095.86	16694.47	21834.98	24297.73	6815.04	26209.42	33993.67
	GGN	P	16200.89	5271.63	17246.74	23276.56	26098.21	8098.30	27917.41	38087.42
	GWN	P	15283.01	5134.79	16414.22	21721.54	23698.88	8072.99	25411.33	33164.72
	GGN	F	16566.91	5597.78	17964.91	23322.77	26791.21	9151.39	29059.91	37468.30
	GWN	F	14915.98	4623.17	16014.96	21466.27	23657.35	7796.18	25309.98	33335.93

Table 4.12: Wind farm $H = 36$ energy output prediction errors comparison

	Model	Graph	Mean Absolute Error [kW]				Root Mean Squared Error [kW]			
			Avg.	$h = 1$	$h = H/2$	$h = H$	Avg.	$h = 1$	$h = H/2$	$h = H$
$W = 1$	MLR	-	24797.87	3365.64	22794.62	28294.66	38215.77	5507.11	35907.93	42651.74
	MFR	-	29240.99	5705.80	28919.38	30507.07	39431.25	9074.50	38069.09	41557.43
	RNN	-	25616.14	11944.53	26626.97	33568.12	38949.54	15735.14	40934.90	50958.85
	TSF	-	24935.54	3743.55	27046.72	31828.14	38787.81	6628.84	42031.75	49024.01
	GGN	P	25188.19	3438.53	27038.20	32588.58	39043.09	5430.74	42130.07	50830.27
	GWN	P	24969.53	7244.51	26956.19	32312.91	38845.71	11161.75	42014.20	49764.91
	GGN	F	25682.54	3188.61	27610.11	33671.78	41478.92	5603.00	44919.63	53493.34
	GWN	F	24821.30	6488.69	26726.86	32609.62	37921.12	10575.78	40513.31	49945.70
$W = 36$	MLR	-	25399.77	7115.05	23136.53	29104.43	39453.50	11106.93	36499.45	44500.77
	MFR	-	28566.54	12146.45	29595.03	30391.74	39759.07	16773.43	38285.68	43398.64
	RNN	-	26806.28	16023.53	27161.10	34548.15	39611.00	20628.71	40703.05	51449.75
	TSF	-	25740.57	7519.32	28059.12	32332.46	39079.45	10134.67	42495.83	49129.88
	GGN	P	25361.49	5849.19	27569.69	32497.54	39927.36	8539.19	43691.80	51183.88
	GWN	P	25027.17	9072.25	26881.97	32505.86	38489.48	15159.92	41209.00	49268.68
	GGN	F	24875.08	4843.32	27108.18	31348.28	39608.36	7634.98	43753.45	50451.89
	GWN	F	24849.12	6807.89	26952.85	32288.08	38484.21	11195.89	41590.78	49630.49
$W = 144$	MLR	-	26924.64	13123.81	25016.10	29779.32	42057.75	19206.00	38841.72	46736.92
	MFR	-	28585.49	19752.95	26292.43	31163.88	41726.42	27067.59	37428.39	45679.62
	RNN	-	27567.10	19073.79	27804.18	34930.89	40520.38	26469.87	41125.56	51476.00
	TSF	-	25121.45	5083.62	27409.71	32219.93	38974.61	8380.28	42285.77	49770.83
	GGN	P	25817.70	6975.38	27748.27	32720.72	40730.40	10360.51	43586.58	51419.56
	GWN	P	24800.76	7463.76	26612.64	32371.75	38705.04	11838.59	41596.39	50034.01
	GGN	F	25390.18	6727.21	27446.37	31915.54	40135.00	10113.94	43315.16	50523.46
	GWN	F	24746.10	6741.60	26811.41	32159.60	38424.59	11296.33	41603.05	49639.64

Table 4.13: Wind farm $H = 144$ energy output prediction errors comparison

Chapter 5

Conclusions

This work investigated the predictive performance of Spatio-Temporal Graph Neural Networks for wind energy forecasting across short-term and long-term predictive horizons, providing an evaluation of their effectiveness through a detailed comparative analysis against alternative architectures. Supported by the spatiotemporal framework introduced in Section 2.2, which serves as the theoretical foundation of this research, the study was conducted on the Spatial Dynamic Wind Power Forecasting dataset. The dataset was explored, prepared, and integrated within this framework, enabling an unbiased comparison with alternative architectures configured to handle the same data and trained under reproducible and consistent conditions. The results of the experiments show that STGNNs consistently outperform competing architectures on both short-term and long-term prediction tasks, demonstrating their potential as a viable solution in the field of wind energy production forecasting.

5.1 Future works

Based on the findings reported in this work, there are several areas for future research on both, the data integration and the modeling sides.

Weather model Wind profiles are characterized by high volatility, and for this reason, the limited set of environmental conditions reported in the SDWPF dataset remains informative for the prediction task only up to a certain horizon, approximately corresponding to six hours. Overcoming this limitation might require incorporating additional environmental features that could provide the system with a more informative internal representation of non-linear weather dynamics, allowing the model to more accurately describe the evolution of wind profiles in the long term.

Temporal relational graph As discussed in Section 4.1, one of the assumptions underlying the experiments reported in this work is the time-invariance of the relational graph $\mathcal{E}$. This assumption implies that the relational information between wind turbines remains static over time. However, this might lead to an underestimation of the reciprocal influence of operational conditions among generators within the wind farm. Introducing this additional granularity could improve predictive accuracy by allowing the model to have a more informative representation of relational dynamics.

Architectural improvements As discussed and visualized in Section 4.4, the models evaluated in the experiments produce a forecast representing the expected energy production profile over the prediction horizon. This approach might lead to an underestimation of the energy output curve variability within that horizon. To overcome this limitation from an architectural perspective, one possible solution is the implementation of an autoregressive decoder, which would allow the model to sequentially generate a more granular forecast for each time step. Accordingly, reformulating the loss function to account for finer-grained energy output fluctuations could potentially lead to improved forecasting accuracy.

Bibliography

[1] International Energy Agency. *World Energy Outlook (2023)*. 2023, p. 353. DOI: "https://doi.org/https://doi.org/10.1787/827374a6-en". URL: https://www.oecd-ilibrary.org/content/publication/827374a6-en (cit. on p. 1).

[2] Albert Betz. *Wind-Energie und ihre Ausnutzung durch Windmühlen*. Aus Naturwissenschaft und Technik. Vandenhoeck & Ruprecht, 1926. URL: https://books.google.lu/books?id=7Lw3AAAAMAAJ (cit. on p. 4).

[3] Jingbo Zhou, Xinjiang Lu, Yixiong Xiao, Jiantao Su, Junfu Lyu, Yanjun Ma, and Dejing Dou. *SDWPF: A Dataset for Spatial Dynamic Wind Power Forecasting Challenge at KDD Cup 2022*. 2022. arXiv: 2208.04360 [cs.LG] (cit. on pp. 6, 15).

[4] J. Meyers, C. Bottasso, K. Dykes, P. Fleming, P. Gebraad, G. Giebel, T. Göçmen, and J.-W. van Wingerden. «Wind farm flow control: prospects and challenges». In: *Wind Energy Science* 7.6 (2022), pp. 2271–2306. DOI: 10.5194/wes-7-2271-2022. URL: https://wes.copernicus.org/articles/7/2271/2022/ (cit. on p. 6).

[5] Florian Ziel, Carsten Croonenbroeck, and Daniel Ambach. «Forecasting wind power – Modeling periodic and non-linear effects under conditional heteroscedasticity». In: *Applied Energy* 177 (2016), pp. 285–297. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2016.05.111. URL: https://www.sciencedirect.com/science/article/pii/S0306261916307164 (cit. on p. 9).

[6] Osamah Basheer Shukur and Muhammad Hisyam Lee. «Daily wind speed forecasting through hybrid KF-ANN model based on ARIMA». In: *Renewable Energy* 76 (2015), pp. 637–647. ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2014.11.084. URL: https://www.sciencedirect.com/science/article/pii/S0960148114008155 (cit. on p. 9).

[7] Yun Wang, Jianzhou Wang, and Xiang Wei. «A hybrid wind speed forecasting model based on phase space reconstruction theory and Markov model: A case study of wind farms in northwest China». In: *Energy* 91 (2015), pp. 556–572. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2015.08.039. URL: https://www.sciencedirect.com/science/article/pii/S0360544215011135 (cit. on p. 9).

[8] Jing Zhao, Zhen-Hai Guo, Zhong-Yue Su, Zhi-Yuan Zhao, Xia Xiao, and Feng Liu. «An improved multi-step forecasting model based on WRF ensembles and creative fuzzy systems for wind speed». In: *Applied Energy* 162 (2016), pp. 808–826. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2015.10.145. URL: https://www.sciencedirect.com/science/article/pii/S0306261915013872 (cit. on p. 9).

[9] Xiaohui Yuan, Chen Chen, Yanbin Yuan, Yuehua Huang, and Qingxiong Tan. «Short-term wind power prediction based on LSSVM–GSA model». In: *Energy Conversion and Management* 101 (2015), pp. 393–401. ISSN: 0196-8904. DOI: https://doi.org/10.1016/j.enconman.2015.05.065. URL: https://www.sciencedirect.com/science/article/pii/S0196890415005300 (cit. on p. 9).

[10] Kuilin Chen and Jie Yu. «Short-term wind speed prediction using an unscented Kalman filter based state-space support vector regression approach». In: *Applied Energy* 113 (2014), pp. 690–705. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2013.08.025. URL: https://www.sciencedirect.com/science/article/pii/S0306261913006600 (cit. on p. 9).

[11] E. Mangalova and E. Agafonov. «Wind power forecasting using the k-nearest neighbors algorithm». In: *International Journal of Forecasting* 30.2 (2014), pp. 402–406. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2013.07.008. URL: https://www.sciencedirect.com/science/article/pii/S0169207013000848 (cit. on p. 9).

[12] A. Lahouar and J. Ben Hadj Slama. «Hour-ahead wind power forecast based on random forests». In: *Renewable Energy* 109 (2017), pp. 529–541. ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2017.03.064. URL: https://www.sciencedirect.com/science/article/pii/S0960148117302550 (cit. on p. 9).

[13] Anwen Zhu, Xiaohui Li, Zhiyong Mo, and Ruaren Wu. «Wind power prediction based on a convolutional neural network». In: *2017 International Conference on Circuits, Devices and Systems (ICCDS)*. 2017, pp. 131–135. DOI: 10.1109/ICCDS.2017.8120465 (cit. on p. 9).

[14] Ruiguo Yu, Jie Gao, Mei Yu, Wenhuan Lu, Tianyi Xu, Mankun Zhao, Jie Zhang, Ruixuan Zhang, and Zhuo Zhang. «LSTM-EFG for wind power forecasting based on sequential correlation features». In: *Future Generation Computer Systems* 93 (2019), pp. 33–42. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2018.09.054. URL: https://www.sciencedirect.com/science/article/pii/S0167739X18314420 (cit. on p. 9).

[15] Qiang Wu, Hongling Zheng, Xiaozhu Guo, and Guangqiang Liu. «Promoting wind energy for sustainable development by precise wind speed prediction based on graph neural networks». In: *Renewable Energy* 199 (2022), pp. 977–992. ISSN: 0960-1481. DOI: https://doi.org/10.1016/j.renene.2022.09.036. URL: https://www.sciencedirect.com/science/article/pii/S0960148122013866 (cit. on p. 10).

[16] Mahdi Khodayar and Jianhui Wang. «Spatio-Temporal Graph Deep Neural Network for Short-Term Wind Speed Forecasting». In: *IEEE Transactions on Sustainable Energy* 10.2 (2019), pp. 670–681. DOI: 10.1109/TSTE.2018.2844102 (cit. on p. 10).

[17] Mei Yu, Zhuo Zhang, Xuewei Li, Jian Yu, Jie Gao, Zhiqiang Liu, Bo You, Xiaoshan Zheng, and Ruiguo Yu. «Superposition Graph Neural Network for offshore wind power prediction». In: *Future Generation Computer Systems* 113 (2020), pp. 145–157. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2020.06.024. URL: https://www.sciencedirect.com/science/article/pii/S0167739X19330626 (cit. on p. 10).

[18] Xiaoxin Pan, Long Wang, Zhongju Wang, and Chao Huang. «Short-term wind speed forecasting based on spatial-temporal graph transformer networks». In: *Energy* 253 (2022), p. 124095. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2022.124095. URL: https://www.sciencedirect.com/science/article/pii/S0360544222009987 (cit. on p. 10).

[19] Andrea Cini, Ivan Marisca, Daniele Zambon, and Cesare Alippi. *Graph Deep Learning for Time Series Forecasting.* 2023. arXiv: 2310.15978 [cs.LG] (cit. on pp. 10–12).

[20] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. *Neural Message Passing for Quantum Chemistry.* 2017. arXiv: 1704.01212 [cs.LG]. URL: https://arxiv.org/abs/1704.01212 (cit. on p. 11).

[21] Ming Jin, Huan Yee Koh, Qingsong Wen, Daniele Zambon, Cesare Alippi, Geoffrey I. Webb, Irwin King, and Shirui Pan. *A Survey on Graph Neural Networks for Time Series: Forecasting, Classification, Imputation, and Anomaly Detection.* 2023. arXiv: 2307.03759 [cs.LG] (cit. on p. 11).

[22] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. «Learning representations by back-propagating errors». In: *Nature* 323 (1986), pp. 533–536. URL: `https://api.semanticscholar.org/CorpusID:205001834` (cit. on p. 38).

[23] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization.* 2017. arXiv: `1412.6980 [cs.LG]`. URL: `https://arxiv.org/abs/1412.6980` (cit. on p. 43).

[24] Sepp Hochreiter and Jürgen Schmidhuber. «Long Short-Term Memory». In: *Neural Computation* 9.8 (1997), pp. 1735–1780. DOI: `10.1162/neco.1997.9.8.1735` (cit. on p. 48).

[25] Victor Garcia Satorras, Syama Sundar Rangapuram, and Tim Januschowski. *Multivariate Time Series Forecasting with Latent Graph Inference.* 2022. arXiv: `2203.03423 [cs.LG]`. URL: `https://arxiv.org/abs/2203.03423` (cit. on p. 52).

[26] Thomas N. Kipf and Max Welling. *Semi-Supervised Classification with Graph Convolutional Networks.* 2017. arXiv: `1609.02907 [cs.LG]`. URL: `https://arxiv.org/abs/1609.02907` (cit. on p. 52).

[27] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. *Graph WaveNet for Deep Spatial-Temporal Graph Modeling.* 2019. arXiv: `1906.00121 [cs.LG]` (cit. on p. 53).

[28] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. *WaveNet: A Generative Model for Raw Audio.* 2016. arXiv: `1609.03499 [cs.SD]`. URL: `https://arxiv.org/abs/1609.03499` (cit. on p. 53).

[29] Sergey Brin and Lawrence Page. «The Anatomy of a Large-Scale Hypertextual Web Search Engine». In: *Computer Networks* 30 (1998), pp. 107–117. URL: `http://www-db.stanford.edu/~backrub/google.html` (cit. on p. 54).