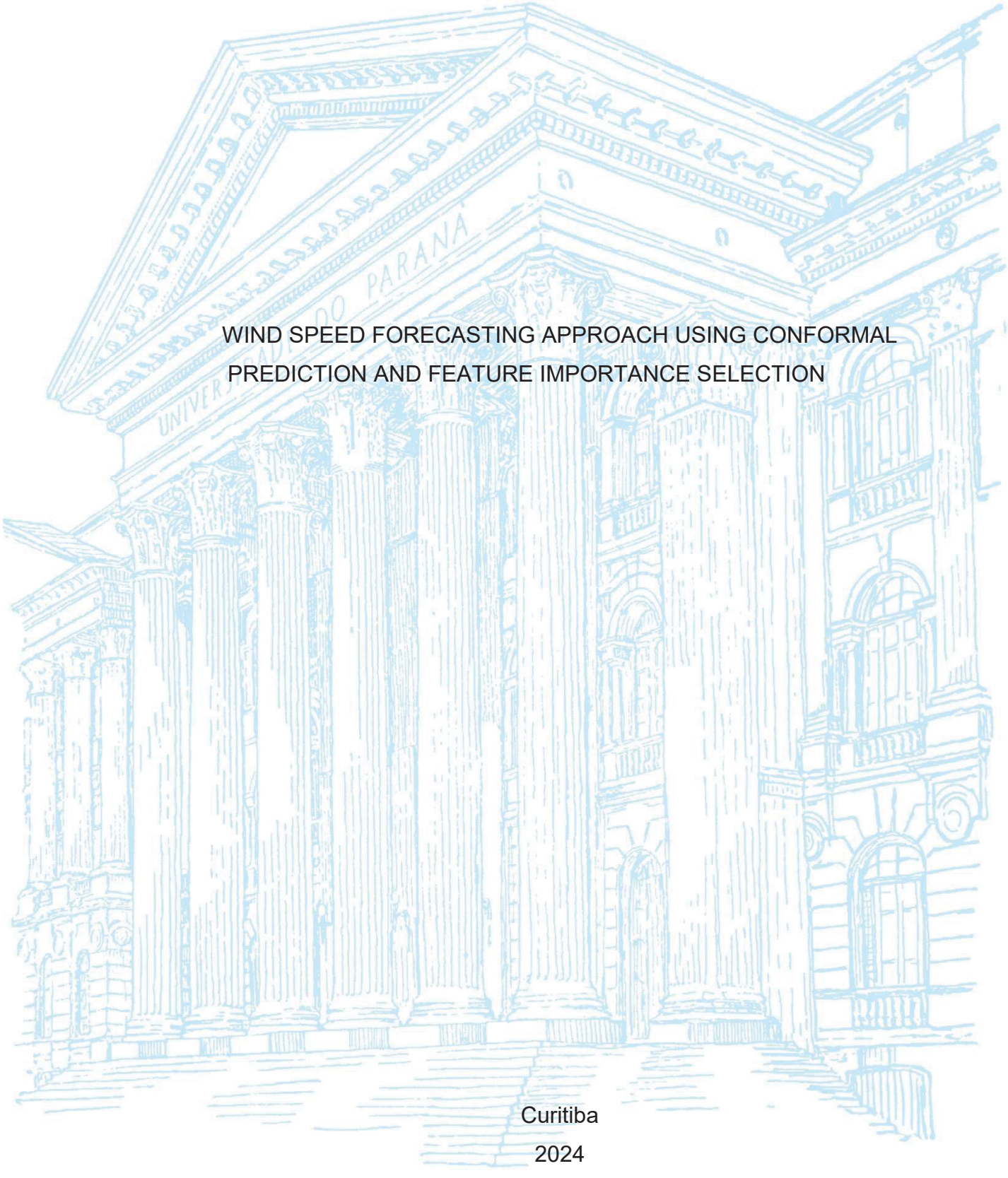
UNIVERSIDADE FEDERAL DO PARANÁ

CESAR VINICIUS ZUEGE PONTE DURA



WIND SPEED FORECASTING APPROACH USING CONFORMAL PREDICTION AND FEATURE IMPORTANCE SELECTION

Curitiba

2024

CESAR VINICIUS ZUEGE PONTE DURA

WIND SPEED FORECASTING APPROACH USING CONFORMAL PREDICTION AND FEATURE IMPORTANCE SELECTION

Dissertação de mestrado apresentada como requisito para a obtenção do grau de Mestre, Setor de Tecnologia, no Programa de Pós-Graduação em Engenharia Elétrica (PPGEE) da Universidade Federal do Paraná.

Orientador: Prof. Dr. Leandro dos Santos Coelho

CURITIBA

2024

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA DE CIÊNCIA E TECNOLOGIA

Dura, Cesar Vinicius Zuege Ponte
Wind speed forecasting approach using conformal prediction and feature importance selection / Cesar Vinicius Zuege Ponte Dura. – Curitiba, 2024.
1 recurso on-line : PDF.

Dissertação (Mestrado) - Universidade Federal do Paraná, Setor de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica.

Orientador: Leandro dos Santos Coelho

1. Aprendizado do computador. 2. Teoria bayesiana de decisão estatística. 3. Energia eólica. 4. Probabilidades. I. Universidade Federal do Paraná. II. Programa de Pós-Graduação em Engenharia Elétrica. III. Coelho, Leandro dos Santos. IV . Título.

Bibliotecário: Leticia Priscila Azevedo de Sousa CRB-9/2029

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação ENGENHARIA ELÉTRICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **CESAR VINICIUS ZUEGE PONTE DURA** intitulada: **WIND SPEED FORECASTING APPROACH USING CONFORMAL PREDICTION AND FEATURE IMPORTANCE SELECTION**, sob orientação do Prof. Dr. LEANDRO DOS SANTOS COELHO, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 14 de Agosto de 2024.

Assinatura Eletrônica
15/08/2024 13:18:48.0
LEANDRO DOS SANTOS COELHO
Presidente da Banca Examinadora

Assinatura Eletrônica
15/08/2024 13:39:31.0
ALEXANDRE RASI AOKI
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
15/08/2024 14:38:39.0
ROBERTO ZANETTI FREIRE
Avaliador Externo (PONTIFICA UNIVERSIDADE CATóLICA DO PARANá)

Assinatura Eletrônica
15/08/2024 14:11:23.0
GIDEON VILLAR LEANDRO
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

RESUMO

Energia eólica é uma fonte de energia renovável em ascensão, desempenhando um papel importante na transição para um sistema energético mais sustentável. A capacidade global total de energia eólica supera 837 GW, evitando mais de 1,2 bilhão de toneladas de dióxido de carbono (CO_2) por ano, e espera-se um ganho de 557 GW na capacidade de parques eólicos até 2026. No entanto, sua variabilidade e a incerteza nas previsões de velocidade do vento apresentam desafios significativos para a geração de energia eólica, afetando sua confiabilidade, eficiência e integração na rede elétrica. A variação na geração de energia eólica é um dos principais desafios enfrentados por essa fonte de energia. Além disso, a incerteza nas previsões de velocidade do vento representa um desafio adicional. As previsões de vento são essenciais para o planejamento e a operação de parques eólicos, mas são complexas devido à natureza dinâmica do vento e à influência de fatores locais.O objetivo geral desta dissertação é propor uma abordagem para previsão de curto prazo de séries temporais com uma medida de incerteza associada à previsão de velocidades do vento, utilizando Predição Conformal, e seleção ótima de variáveis com base em sua importância, utilizando Shapley, para melhorar a compreensão das variáveis e resultados, fornecendo uma variável de entrada mais confiável e previsível para modelos de previsão de potência de turbinas eólicas ou outras aplicações. Esta dissertação utiliza os modelos propostos em dois casos relacionados à previsão de velocidade do vento: Beutenberg (Alemanha) e Limoeiro (Brasil).Este trabalho constatou que, para ambos os casos, a decomposição de séries temporais usando Decomposição Modal Variacional (VMD), aliada à Análise de Espectro Singular (SSA), para alimentar um modelo de predição conformal melhorou o desempenho do modelo. No caso um, o modelo campeão foi a Máquina de Gradiente Boosting Leve (LGBM)+VMD+SSA sem ajuste parcial, resultando em um erro quadrático médio (RMSE) de 0,251, cobertura de 94,4% e largura de 1,008. No caso dois, o melhor modelo foi o LGBM+VMD+SSA sem ajuste parcial, resultando em um RMSE de 0,21597, cobertura de 90,3% e largura de 0,678. Além disso, os modelos propostos neste trabalho podem ser utilizados para prever a velocidade do vento e a geração de energia eólica.

Palavras-Chave: Previsão Conformal, Previsão Probabilística, Aprendizado de Máquina, Otimização Bayesiana, Previsão de series temporais

ABSTRACT

Wind power is a renewable energy source on the rise, playing an important role in transitioning to a more sustainable energy system. Total global wind energy capacity stands at more than 837 GW and avoids more than 1.2 tons of carbon dioxide (CO_2) per year, and a gain of 557 GW in wind farm capacity is expected by 2026. However, its variability and the uncertainty in wind speed forecasts present significant challenges for wind power generation, affecting its reliability, efficiency, and integration into the electricity grid. The variation in wind power generation is one of the main challenges faced by this energy source. Furthermore, the uncertainty in wind speed forecasts represents an additional challenge. Wind forecasts are essential for planning and operating wind farms but are complex due to the dynamic nature of the wind and the influence of local factors. The general objective of this dissertation is to propose an approach for short-term forecasting of time series with a measure of uncertainty associated with forecasting wind speeds, using Conformal Prediction, and optimal selection of features based on their importance, using Shapley, to improve the understanding of features and results while providing a more reliable and predictable input variable for wind turbine power forecasting models or other applications. This dissertation uses the proposed models in two cases regarding wind speed forecasting: Beutenberg (Germany) and Limoeiro (Brazil). This work found that for both cases the decomposition of time series using Variational Mode Decomposition (VMD) allied with Singular Spectrum Analysis (SSA) to feed into a conformal prediction model improved the performance of the model, for case one the champion model was Light Gradient Boosting Machine (LGBM)+VMD+SSA without partial fit, resulting in a root mean squared error (RMSE) of 0.251, coverage of 94.4% and width of 1.008. For case two, the best model was the LGBM+VMD+SSA without partial fit resulting in an RMSE of 0.21597, a coverage of 90.3%, and a width of 0.678. Also, the models proposed in this work can be used to predict the wind speed and the wind power generation.

Keywords: Conformal Prediction, Probabilistic Forecasting, Machine Learning, Bayesian Optimization, Time Series Forecasting.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

Abbreviations	Meaning
AI	Artificial Intelligence
AM-FM	Amplitude Modulation-Frequency Modulation
ANN	Artificial Neural Network
AR	Auto-Regressive
AR + PCR + KF	AutoRegressive + Principal Component Regression + Kalman Filter
ARFIMA-APARCH	AutoRegressive Fractionally Integrated Moving Average - Asymmetric Power AutoRegressive Conditional Heteroskedasticity
ARIMA	Auto-Regressive Integrated Moving Average
ARMA	Auto Regressive Moving Average
BP	Back Propagation
CCP	Cross Conformal Prediction
CE	Complete ensemble
CLSTM	Contextual LSTM
CNN	Convolutional Neural Networks
CP	Conformal Prediction
CQR	Conformized Quantile Regression
CRF	Conformized Random Forest
CS	Cuckoo search
CSVM	Conformized Support Vector Machine
DeepLIFT	Deep Learning Important Features
DRNN	Deep Recurrent Neural Network
DT	Decision Tree
Enbpi	Ensemble batch prediction intervals
ERNN	Elman neural network
FIR	Finite Impulse Response
G	Grey Correlation
GBM	Gradient Boosting Machine
GM	Grey forecasting method
GRAD-CAM	GRADient Class Activation Mapping
GRNN	Generalized regression neural network
GWPPT	Generalized Wavelet Packet Transform
ICP	Inductive Conformal Prediction
IIR	Infinite impulse response
KF	Kalman Filter
KFNN	Kalman Filter Artificial Neural Network

LAF-MLN	Locally Adaptive Feedforward - Multi-Layer Network
Lasso	Least Absolute Shrinkage and Selection Operator
LIGHTGBM	Light Gradient Boosting Machine
LIME	Local Interpretable Model-agnostic Explanations
LOESS	Locally Estimated Scatterplot Smoothing
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MAPIE	Margin-based Adaptive Prediction Interval Estimation
MLP	Multi-Layer Perceptron
MWS	Multi-Objective WDO with Simulated Annealing
Mycielski	Mycielski
NARX	Nonlinear AutoRegressive with eXogenous inputs
NP Regression	Nonparametric Regression
PAERNN	Partially Adaptive Elman Recurrent Neural Network
PERNN	Partially, the Elman Recurrent Neural Network
PMERNN	Partially Modified Elman Recurrent Neural Network
PSO	Particle Swarm Optimisation
R2	Coefficient of Determination
RF	Random Forests
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RSTD	Regime-switching space-time diurnal
SDA + SVR + UKF	Stacked Denoising Autoencoder + Support Vector Regression + Unscented Kalman Filter
SEATS	Seasonal Extraction ARIMA Time Series
SHAP	SHapley Additive exPlanation
sMAPE	Symmetrical Mean Average Percentage Error
SMNDE	Spatiotemporal multi-network deep ensemble
SSA	Singular Spectrum Analysis
STACK	Stacking ensemble method
static MLP	Static Multi-Layer Perceptron
STLD	Seasonal Trend LOESS Decomposition
SVARX-TARCHX	Structural Vector AutoRegressive with eXogenous inputs - Threshold AutoRegressive Conditional Heteroskedasticity with eXogenous inputs
SVD	Singular Values Decomposition
SVM	Support Vector Machines

TCP	Transductive Conformal Prediction
TDD	Trigonometric Direction Diurnal
VAR	Vector AutoRegressive
VAR-TARCH	Vector AutoRegressive Threshold AutoRegressive Conditional Heteroskedasticity
VMD	Variational Mode Decomposition
v-SVM	v-support vector machine
WDO	Wind Driven Optimization
WPF	Wind Power Forecasting
WPPT	Wavelet Packet Transform
WSF	Wind Speed Forecasting
XAI	eXplainable Artificial Intelligence

1 INTRODUCTION

This chapter is organized as follows. Section 1.1 presents the contextualization of this project and problematization, and section 1.2 presents the general and specific objectives of this work.

1.1 CONTEXTUALIZATION

Wind power is a renewable energy source on the rise, playing an important role in transitioning to a more sustainable energy system. Total global wind energy capacity stands at more than 837 GW and avoids more than 1.2 tons of carbon dioxide (CO_2) per year, and a gain of 557 GW in wind farm capacity is expected by 2026 (GWEC, 2022). However, its intermittent and the uncertainty in wind speed forecasts present significant challenges for wind power generation, affecting its reliability, efficiency, and integration into the electricity grid.

There are positive points associated with wind energy. Firstly, wind power is a clean and renewable energy source, contributing to the reduction of greenhouse gas emissions and the mitigation of climate change (IEA Wind Electricity, 2022). In addition, wind technology has developed rapidly, resulting in more efficient turbines and a reduction in generation costs (IRNEA, 2021). This makes wind energy increasingly competitive with conventional energy sources. Another major point in favor of the accelerated growth predicted for wind power generation is the commitment to the scenario of zero net emissions of CO_2 by 2050, where capacity is expected to quadruple by the end of the decade (IEA Wind Electricity, 2022).

Another positive point is the diversification of the energy matrix. Wind energy reduces dependence on fossil fuels, helping to increase energy security and reduce the volatility of energy prices (European Commission, 2020). In addition, the wind industry boosts local economic development, creating jobs and attracting investment in rural and coastal areas (IRNEA, 2021).

The variation in wind power generation is one of the main challenges faced by this energy source. The generation of electricity from wind is affected by the variability of wind conditions over time, resulting in fluctuations in energy production (GWEC,

2022), from moments of high generation, when the wind is strong, to periods of low or no generation when the wind is weak or absent.

Furthermore, the uncertainty in wind speed forecasts represents an additional challenge. Wind forecasts are essential for planning and operating wind farms but are complex due to the dynamic nature of the wind and the influence of local factors (DNV, 2022). Uncertainties in these forecasts can lead to errors in the estimation of wind generation, affecting the reliability of the energy supply.

It is also important to consider the design factors that directly affect wind power generation. Proper site selection, based on wind speed and consistency, is key (DNV, 2022). Accurate assessment of the wind resource, through detailed measurements and analysis, is essential to determine the viability of the project (DNV, 2022). The selection of appropriate wind turbines, considering generation capacity, hub height, rotor diameter, and wind turbine efficiency, is also crucial.

Multi-horizon wind speed forecasts are used for planning the operation of wind farms, managing grid integration, optimizing wind turbine efficiency, market clearing, and equipment maintenance.

1.2 OBJETIVES

The general objective of this dissertation is to propose an approach for short-term forecasting of time series with a measure of uncertainty associated with forecasting wind speeds and optimal selection of features (characteristics or attributes) based on their importance in order to improve the understanding of features and results and provide a more reliable and predictable input variable for wind turbine power forecasting models, in both cases used in this study.

Therefore, the specific objectives of this dissertation are:

- Related work study of the use of uncertainty measures associated with time series forecasting and the explanation of variables in a model.
- Exploratory data analysis and study of dimensionality reduction and its effects on short- and long-term forecasting of wind speed time series.

- Tests of machine learning candidate models such as Random Forest, Light Gradient Boosting Machine (LightGBM), Decision Tree, and Kalman Filter Artificial Neural Network (KFNN).
- Evaluation of different datasets decompositions and results analysis such as Variational Mode Decomposition, Singular Spectral Analysis, and Seasonal Decomposition.
- Provide a measure of uncertainty associated with wind speed forecasts linked to the use of Conformal Prediction.
- Interpret and filter system variables based on their importance using Explainable Artificial Intelligence (XAI) methods.
- Refine Hyperparameters with Cross-validation and Optuna Framework and check model performance with Root Mean Squared Error (RMSE), Symmetrical Mean Average Percentage Error (sMAPE), and coefficient of determination (R^2).

This study has two major contributions validated by presenting two case studies. The first major contribution is devoted to blend conformal prediction on wind speed forecasting which is a key metric to forecast the power generated by wind turbines and pivot to other electric source generation in the short term to keep the network running. The second major contribution evaluates the efficiency of multiple signal decompositions on wind speed forecasting area and its best combinations on short-term forecasting. The third contribution lies in the use of probabilistic forecasting to deliver a statistically trusted set of wind speed forecasts in the short term, bringing more trust and usability to the model in real scenarios where the point forecasting error affects drastically the network.

The remainder of this dissertation is organized as follows. Chapter 2 contains a theoretical foundation on the subjects covered in this work. Chapter 3 is a brief literature review of published works and articles related to the dissertation and the techniques that will be used. Chapter 4 presents each case study, its origins, and details regarding the main forecasting purpose. Chapter 5 aims to break each case into two new topics to discuss the results, outputs of each model, decompositions, and probabilistic forecasting. Finally, chapter 6 states the findings of this study and discusses open topics for further studies.

2 THEORETICAL BACKGROUND

In this chapter, the forecasting models employed in each case study described in chapter 4 are detailed.

2.1 TIME SERIES

Time series are a sequence of observations sampled sequentially and equally spaced in time (Box et al., 2015). In other words, they are data of any nature with strong and explicit time dependence between the samples, as shown in Table 1.

Time (min)	Wind Speed (m/s²)
0	0
1	2.5
...	...
t	*X*

Table 1 - Time Series Example

Source: Author, 2024

The main interest is in a prior observation made at lag times to get a base to test, prepare, train, and retest the forecast method chosen or just describe the time series. The reference overtime is the ‘zero’ on the time axis and this is equal to the current time, so, all previous values must be negative by referring to the current time and the future values must be positive by logic.

A standard term used for time is t , and the observation typically is y , so $y(t)$, refers to the current data observation, $y(t-1)$ is the last observation, and $\hat{y}(t+1)$ is the one future value forecasted? As a simplification, often the $y(t)$ is dropped from the notation resulting to t as the current time and point of reference, $t-n$ a lag time by n and $t+n$ a forecasted time.

Forecasting involves taking models to fit historical data and using them to predict future observations (BROWNLEE, 2016).

Time series often have intrinsic components such as Trend (T), Seasonality (S), Cycle (C), and Noise (Er). These components can be acquired using various techniques with the aim of better understanding the signal input, modeling each component separately, and facilitating data forecasting. A time series can be decomposed, but the decomposition doesn't need to include all the components.

Trend (T): This is the component that carries a long-term upward or downward trend in the data, which can be linear or non-linear, and trend changes can occur.

Seasonality (S): This is the component that carries information on seasonal factors that influence the data, such as daily, weekly, monthly periods, and so on.

Cycle (C): This is the component that carries information on fluctuations in the variable's values lasting more than a year. This component is also usually used in conjunction with the trend component, resulting in a single component called Trend-Cycle.

Noise (Er): This is the component that will carry all the information about unexplained fluctuations of a random nature.

2.1.1 TIME SERIES DECOMPOSITION

Classical Seasonal Decomposition (Persons, 1919) can be broken down into two methods, additive and multiplicative. This method is done using centered moving averages (CMA) and aims at approximation, not forecasting, but it is advantageous to deseasonalize the observation allowing the model to focus on predicting the general trend of the data. Also, the seasonal component can provide much more information regarding the behavior of that observation, allowing new possibilities for analysis and prediction.

In Classical decomposition, the trend equation varies if the seasonal period m *is* even ($m = 4$ for quarterly, $m = 12$ for monthly) or odd ($m = 7$ for daily). The de-trend $\hat{T}$ is a moving average (MA) process of $(k \cdot m)$-MA where $k = 1 \implies \frac{m}{2} \notin \mathbb{Z}$ and $k = 2 \implies m/2 \ni \mathbb{Z}$ lefting a signal with seasonal and noise component. The de-seasonal $\hat{S}$ is estimated as na average of the detrended series for that season of

successive years (S^1 is the average across all Januaries, and so on) and all left is residual component $\widehat{Er}$.

Decomposition by the additive model is used when the components do not vary with the magnitude of the data and can be described as equation 2.1.1. Decomposition by the multiplicative model is used when the components vary with the proportional magnitude of the data and can be described as equation 2.1.2. Identifying an additive or multiplicative behavior can be tricky since oftentimes one component might be additive while others are multiplicative, but a simple rule can be tested: if the magnitude of seasonal component changes over time, then the time series has multiplicative behavior.

$$y(t) = T + S + C + Er \tag{2.1.1}$$

$$y(t) = T \cdot S \cdot C \cdot Er \tag{2.1.2}$$

Regarding the Classical decomposition method, the estimate of the trend is unavailable for the first and last few observations (if $m = 12$ there will not have estimation for the first and last six observations), the seasonal component repeats from year to year and is not a realistic scenario and is not robust handling with outliers since the algorithm lies on averages.

It is important to note that the noise component Er is not included in the forecasting model because it is random and theoretically unpredictable. To illustrate the concept of decomposing a time series, Figure 2.1 shows a case of an energy demand time series sampled every 1 hour and its respective components.

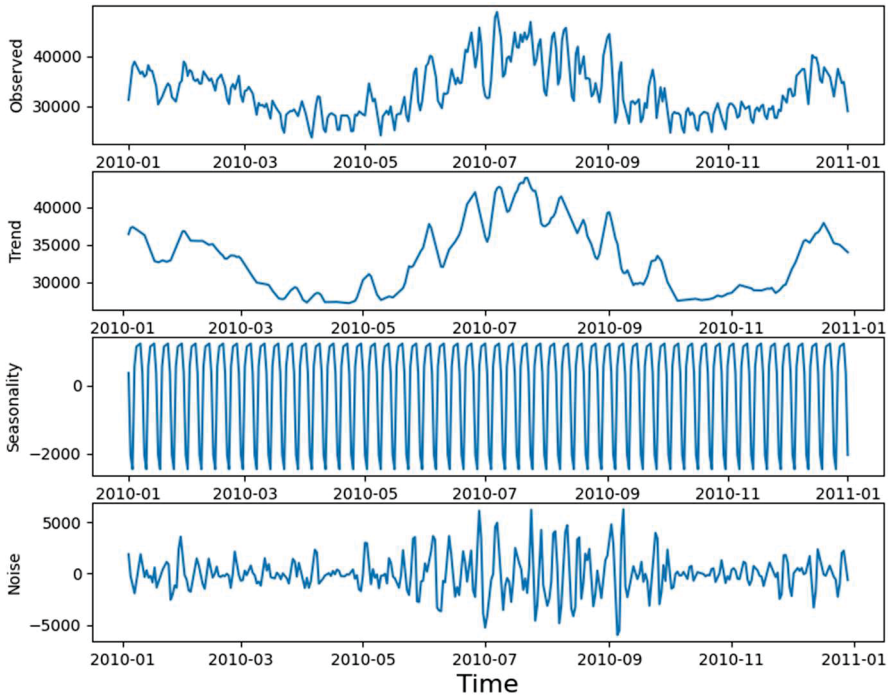


Figure 2.1 - Time Series Decomposition with Classical Method on Electric Power Dataset

Source: Author, 2024

2.1.1.1 X-11

The X-11 method was originated in the US (United States) Census Bureau in 1957 and further developed by Statistics Canada. This technique overcomes the classical method in some weaknesses where trend-cycle estimation is available for all observations, trend handles sophisticatedly on day variations and cyclical or known effects like Holliday, and last but not least seasonal component can vary slowly over time.

This method accepts additive or multiplicative decomposition like the classical method, but it handles internally the de-trend and de-seasonal process dynamically and the noise or error is called “irregular”.

On the other hand, X-11 only handles monthly and quarterly data, with no prediction or confidence intervals about the results because of the iterate algorithm built.

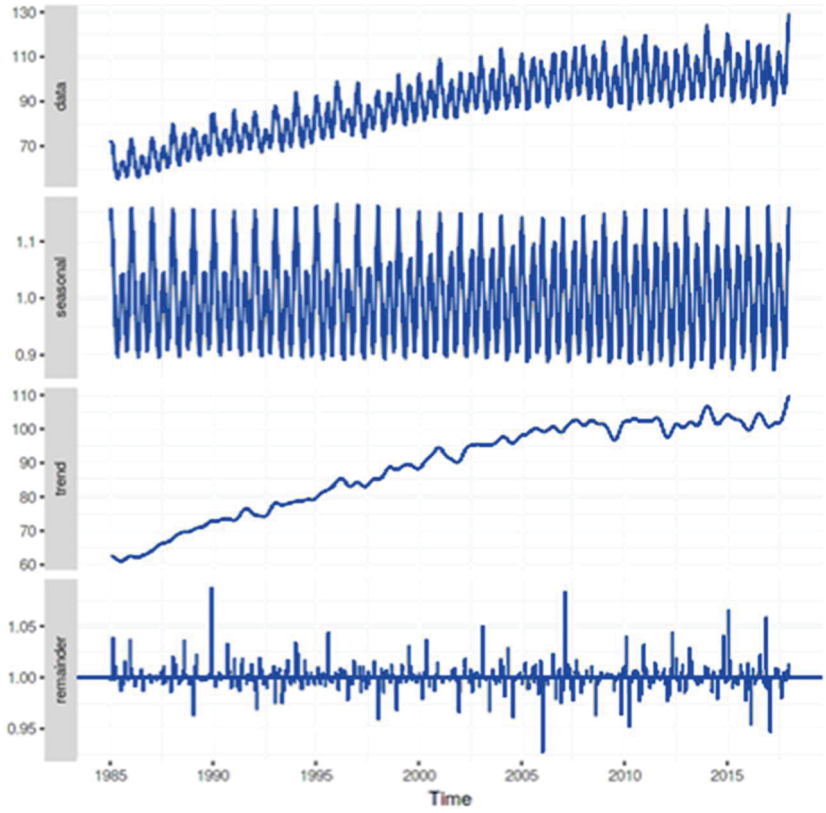


Figure 2.2 - Time Series Decomposition with X-11 on electrical equipment index dataset

Source: Author, 2024

2.1.1.2 SEASONAL AND TREND DECOMPOSITION USING LOESS

Seasonal and trend decomposition using the locally estimated scatterplot smoothing (STL) method was developed by (Cleveland et al., 1990) to overcome the drawbacks of earlier algorithms like X-11 and SEATS. A huge enhancement of this algorithm over X-11 lies in being applied to every type of seasonality, not only monthly or quarterly data. Also, the seasonal component can change over time and the rate of change can be controlled by the user, and trend-cycle smoothness as well. Last but not least, it can be robust regarding outliers once the user can specify a robust decomposition affecting the noise component. An example of STLD is shown in Figure 2.3.

Once life is not a bed of roses, STL has some disadvantages as well. STL does not handle trading day or calendar adjustments automatically and accepts only additive decomposition. Multiplicative decomposition could be obtained with STL taking logs and then back-transforming the components to get multiplicative

decompositions. Box-Cox transformation can also be used to achieve decompositions that are between additive and multiplicative.

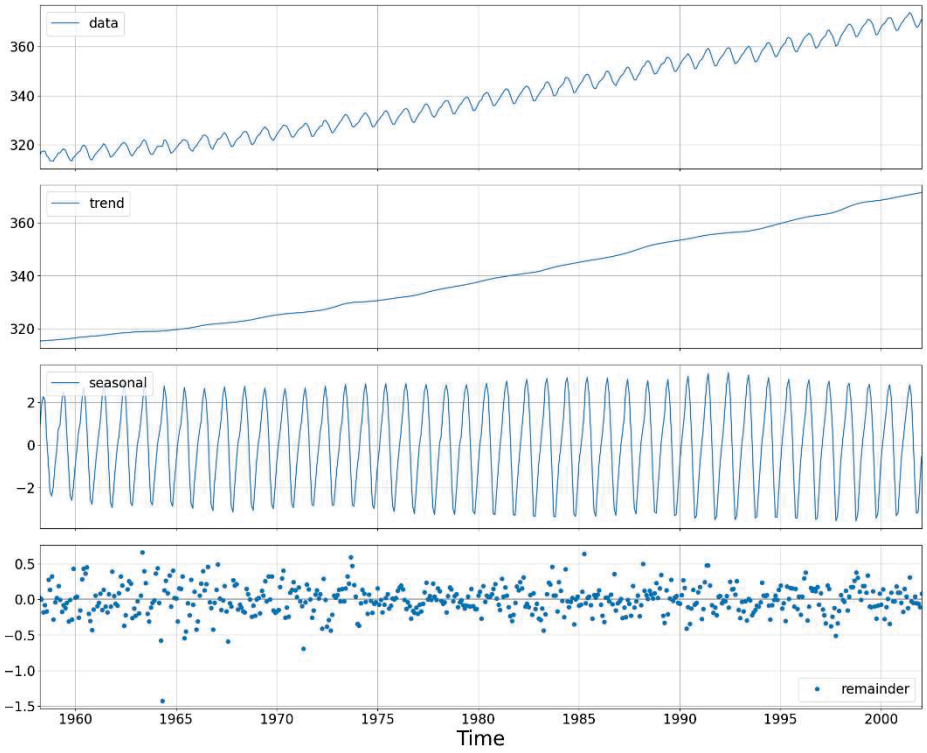


Figure 2.3 - Time Series Decomposition with STL on CO_2 dataset. Values in ppmv

Source: Author, 2024

2.1.1.3 SINGULAR SPECTRUM ANALYSIS

The Singular Spectrum Analysis (SSA) method was developed by Broomhead and King (1986) but got very known after Golyndina et al., (2001). SSA is an analysis and prediction technique that aims to decompose the original signal into a sum of small numbers of interpretable components such as trend, seasonality, cycle, other oscillatory components, and noise.

Behind the scenes, SSA is a group of other processes or steps, that combined have this power, but there is no need for parametrical models, stationarity assumptions, or uncorrelated noise (Golyandina et al., 2001). Summing up, SSA has four steps: embedding, singular value decomposition (SVD), grouping, and diagonal averaging which will not be detailed in this work.

Given a wind speed time series, SSA needs a number L of decompositions, or window length, to run the code behind the scenes where L is limited to be lower than half of the size of the dataset, let us take $L = 20$ which decomposition can be seen in Figure 2.4. As presented before, some smoother components and cyclic components are split, as the noise, so the components can have a high correlation with nearby components that should be aggregated for the sake of simplicity and optimization of further steps on prediction, for example. The weighted correlation information is presented in Figure 2.5.

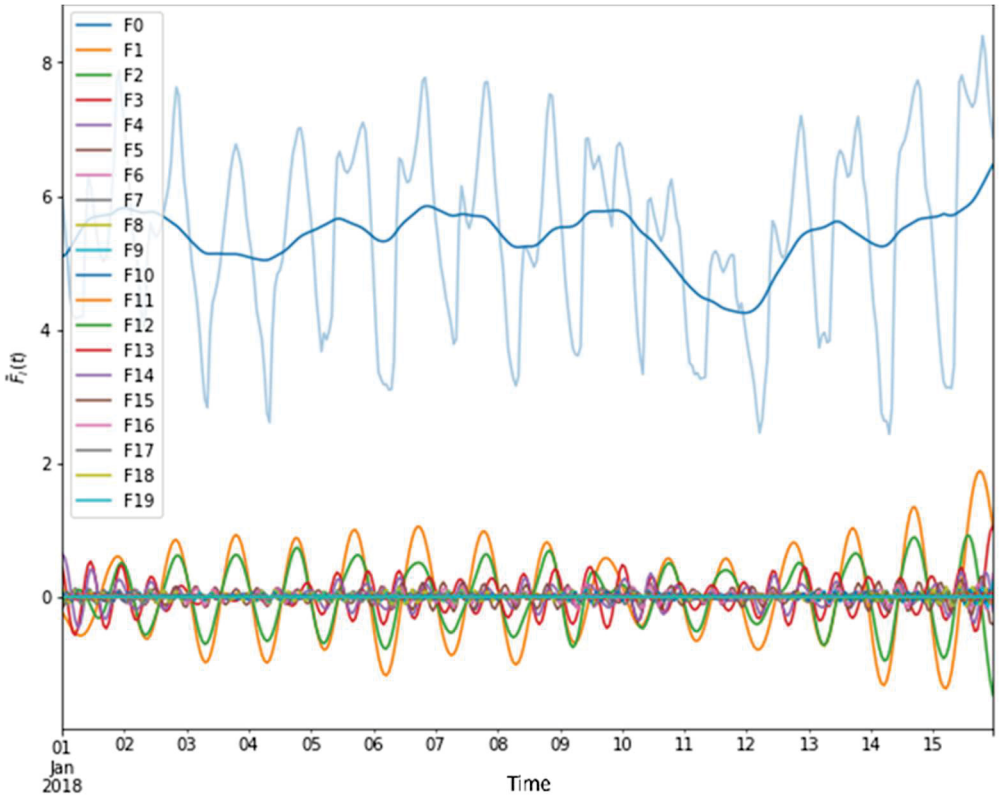


Figure 2.4 - Time Series Decomposition with SSA on Wind Speed dataset

Source: Author, 2024

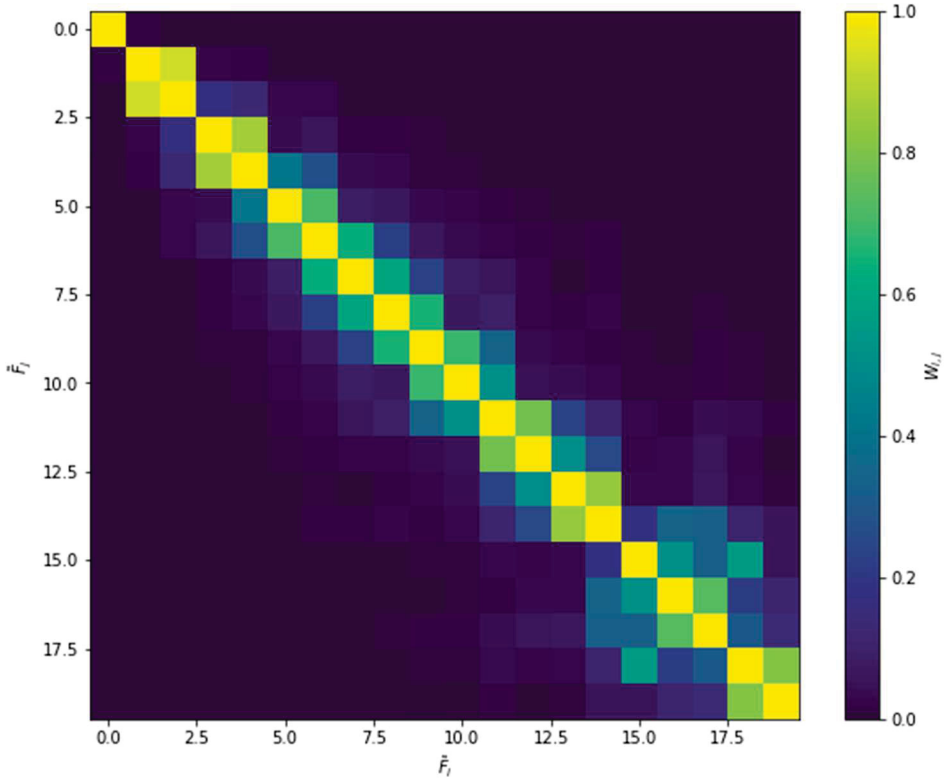


Figure 2.5 - Weighted Correlation Matrix of Wind Speed using SSA

Source: Author, 2023

Based on Figure 2.5, the components as 1 and 2 should be grouped as 3 with 4 but in this case, there's a high correlation (>60%) between 4, 5, and 6, this case should be analyzed carefully because a more aggregated version is better but the complexity of the resultant component could not be easily predicted or may affect the signal with harmonics or noise and the same should be done until the end, always remembering that the last components tend to be noise but the question is: "Which component does the noise start at?".

After all analysis and grouping of components, there must be a lowest K chosen components as possible that better describe the signal without the noise and other useless components, in this process metrics such as Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE) can help to better detect the effectiveness of aggrouped components K_i in the reconstructed signal over the true signal. The result of the reconstructed signal, noise, and the true signal on the wind speed dataset can be observed in Figure 2.6.

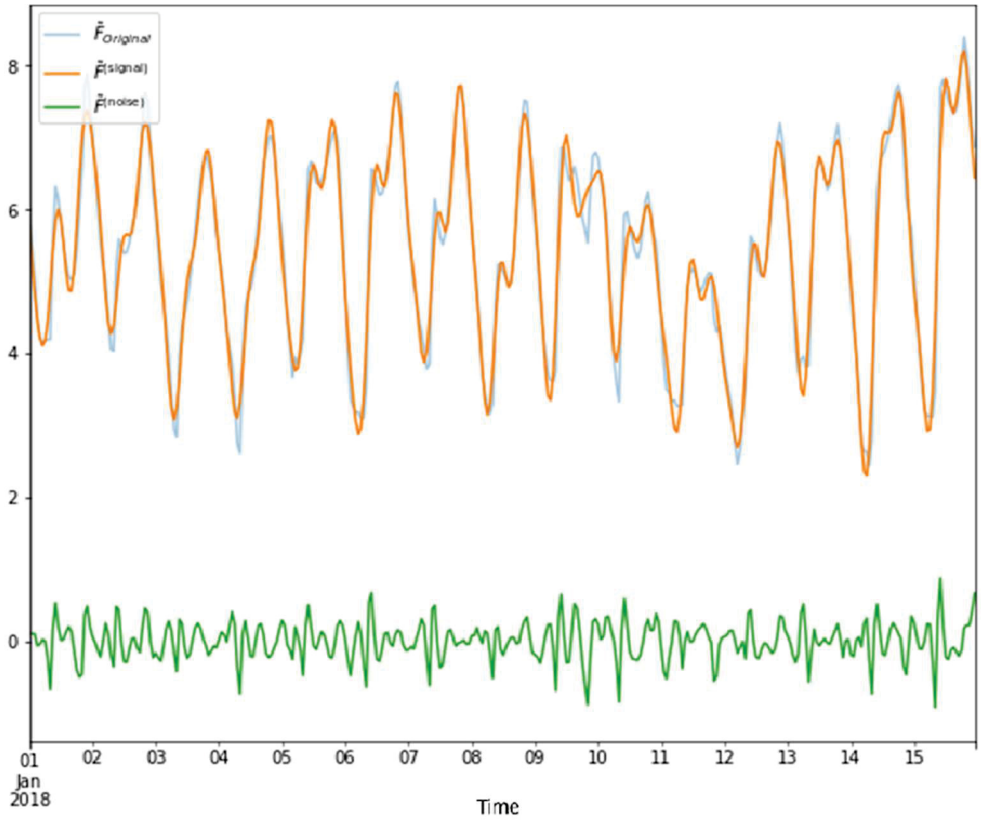


Figure 2.6 - Reconstructed Signal (Orange) Versus Original (Light Blue) and Noise (Green) of Wind Speed using SSA

Source: Author, 2024

2.1.1.4 VARIATIONAL MODE DECOMPOSITION

The variational mode decomposition (VMD) is an adaptive and non-recursive technique applied to signal decomposition problems. The technique was proposed by Dragomiretskiy and Zosso (2014). Given a signal, VMD aims to decompose it in multiple sets of pulse modes u_k bandwidth-limited, Intrinsic mode functions (IMFs), where each mode is compressed around the pulse central frequency w_k.

IMFs or sub-signals, have a high dependency on the bandwidth for the decomposition process. There are different ways to compute this bandwidth such Hilbert transform on each mode u_k (Moreno et al., 2019), exponential signal mixture with estimated frequencies and Gaussian smoothing of demodulated signal to reconstruction.

VMD can underperform and lead to miss decomposition when the sampling frequency is near the Nyquist theorem limit. Overall, VMD effectively avoids the propagation of errors resulting from the traditional recursive filtering procedure present in empirical mode decomposition (EMD) and therefore achieves more accurate results in the reconstruction of the components. Also, the total removal of noise from a signal can be achieved by eliminating the Lagrange multiplier from the algorithm, as it is responsible for ensuring the fidelity of the reconstructed signal to the original signal.

SSA, as seen previously, has the problem of not guaranteeing the separability of the decomposed signal, which compromises the prediction, which is not the case with VMD once this technique does not use concepts from linear algebra where the orthogonality criteria must be satisfied. But keep in mind that both techniques can be ensemble to achieve an even better decomposition such VMD-SSA hybrid technique.

As an example of pure VMD, given a hypothetical data set with known frequencies of 5Hz, 70Hz, and 250Hz with a random noise as shown in Figure 2.7, the dataset must be divided into k IMFs such they are centered on these frequencies. The value of k in this case, is easy to obtain once the frequencies and their number are known. So, $k = 3$ for this case. In a real-world scenario, the number of modes k is the first barrier, to circumvent this, a brute force approach or a more robust frequency domain algorithms is to analyze it such as AM-FM demodulation.

Like the mode, VMD has a high dependency on bandwidth because of the Wiener filter, so this is the next hyperparameter to be set. Reducing the value of α increases the bandwidth of the Wiener filter, resulting in an increase in the bandwidth around the center frequency in the filtered signal. This allows to extract components with high fidelity in relation to the original signal, but highly correlated with remanescent components, which is not what is wanted for the prediction of each mode. The actual value is defined by the problem and by the other parts of the pipeline of prediction or decomposition. Draomiretskiy and Zosso (2014) set a maximum value of, in case of modes reconstruction with minimum error with the original signal, but in this work, a time series is used and is not perfectly bandwidth limited and the goal is to achieve an estimation of frequencies of the signal.

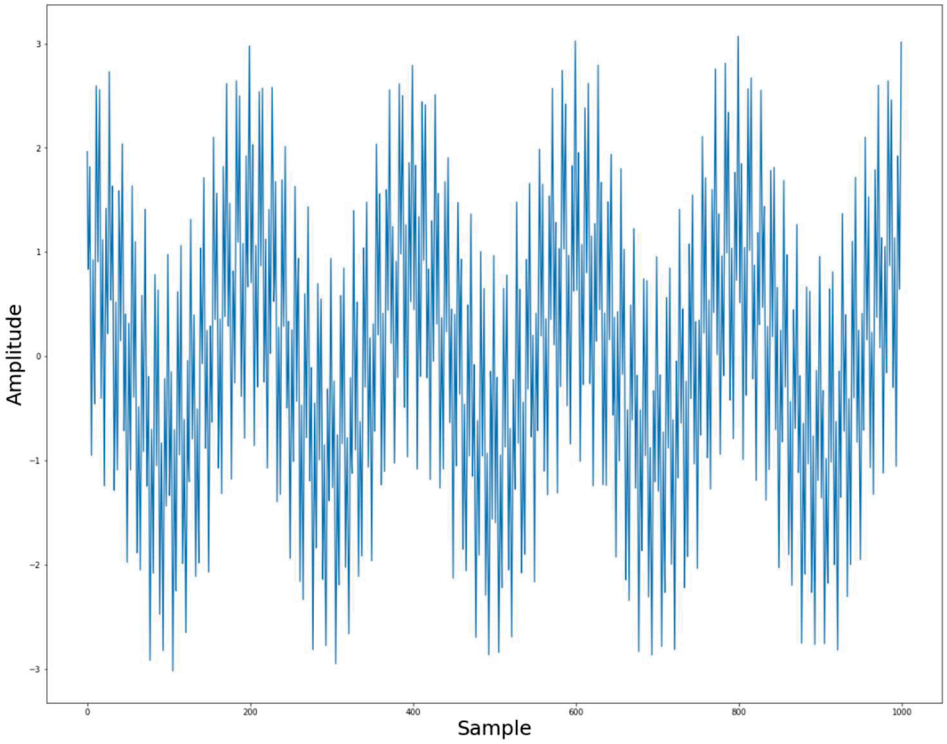


Figure 2.7 - Signal composed by frequencies of 5Hz, 70Hz and 250Hz

Source: Do Autor, 2023

In the example case, using $k = 3$, $\alpha = 500$, the frequencies achieved were extremely near to the original frequencies, as can be seen in Figures 2.8, 2.9, and 2.10.

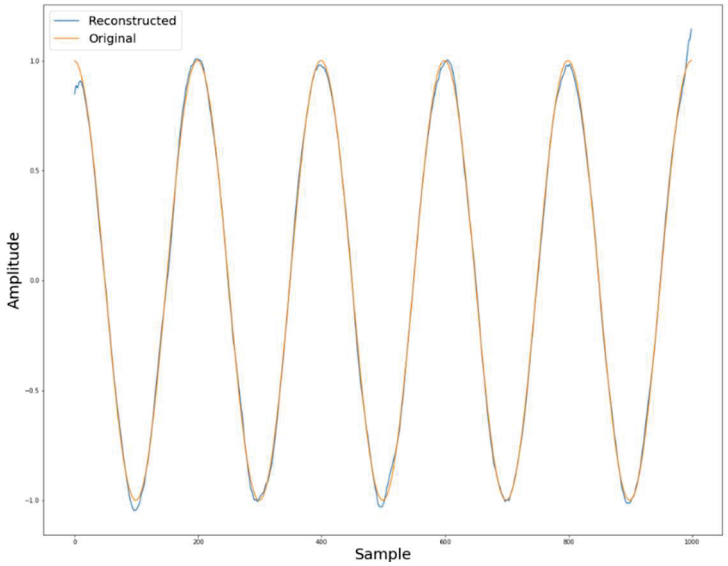


Figure 2.8 - Signal reconstruction vs original mode of 5Hz

Source: Author, 2024

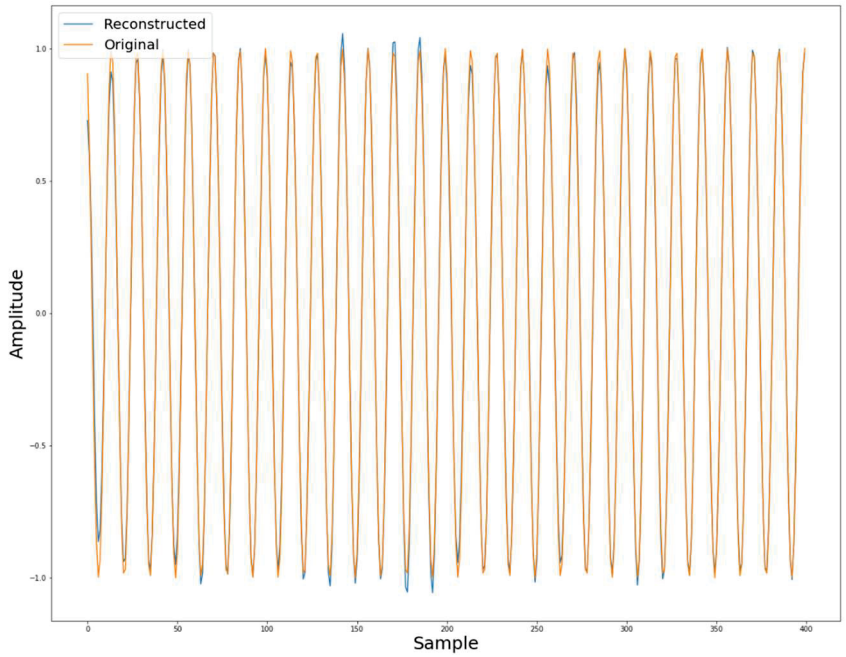


Figure 2.9 - Signal reconstruction vs original mode of 70Hz

Source: Author, 2024

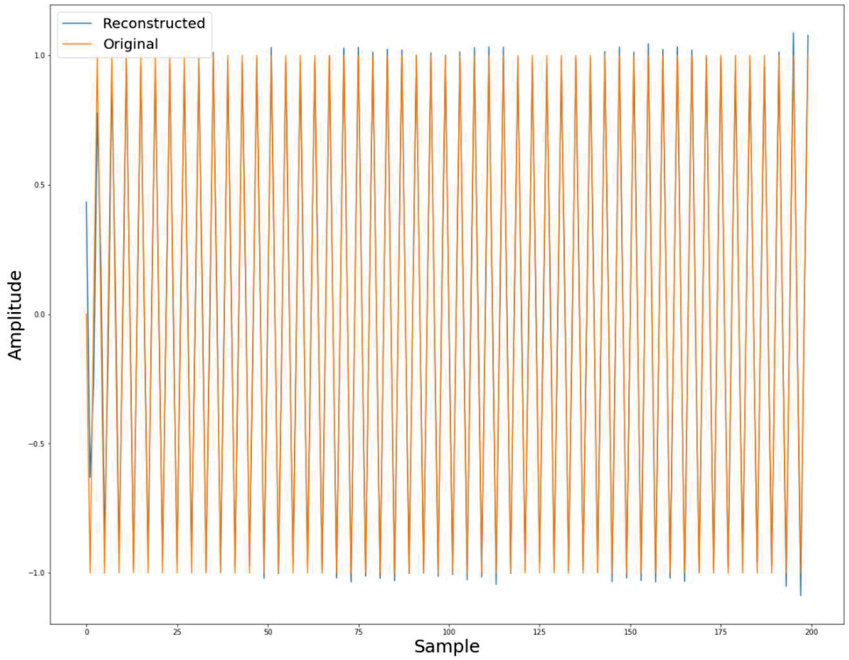


Figure 2.10 - Signal reconstruction vs original mode of 250Hz

Source: Author, 2024

2.2 MACHINE LEARNING

Machine learning (ML) is a branch of artificial intelligence that allows computer systems to learn and improve from data, without the need to be explicitly programmed. There are categories within the field of machine learning: Supervised Learning, Unsupervised Learning, and Reinforcement Learning where each category is used for a different purpose, as shown in Figure 2.11.

Supervised Learning is a group of machine learning algorithms that require both input and output variables to be provided. This data will be used to train the ML model which results in a function that best represents that data and can be used to predict future data, whether continuous (Regression) or discrete (Classification).

Unsupervised Learning is the group of algorithms that use input data without labels defined for the output. The main objective is to distribute the data into categories so that the result is more informative compared to the input data.

Reinforcement Learning is a group of algorithms that aim to achieve the highest numerical reward for a given action by balancing the parameters of state, reward, and rules of the environment, thus creating a dynamic environment. A state is the numerical information that feeds the agent, which is stored and used to make a certain decision. The result of the action is given a numerical value called a reward, which can be positive or negative. The reward feeds back to the agent so that it can readjust the parameters and carry out new actions.

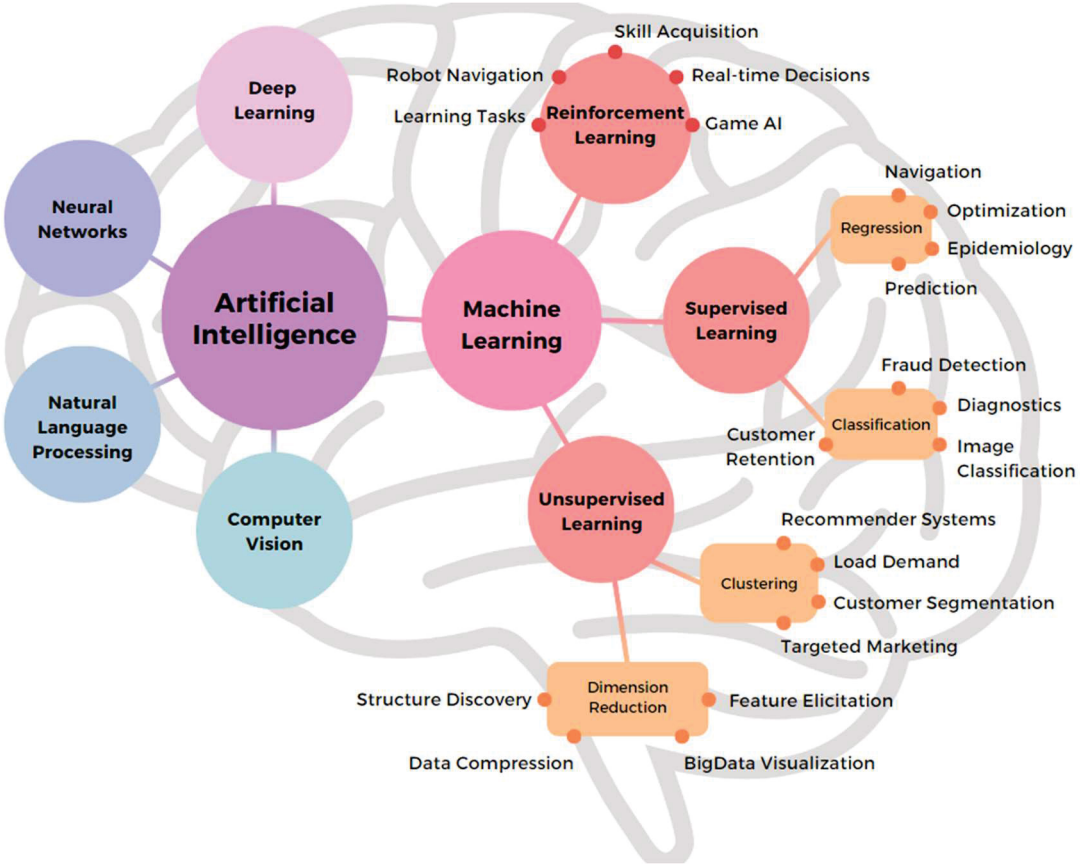


Figure 2.11 – Some AI Fields

Source: Author, 2024

2.2.1 DECISION TREE

Tree-based algorithms that manage multi-output issues with minimal data preprocessing are referred to as decision trees. It functions similarly to a map in order to determine the likely outcomes of several connected decisions. Generating a model that learns simple decision rules derived from the attributes of the data is the aim. Typically, the process begins with a single node and branches into likely outcomes. Every one of those result nodes then contributes to other ones that are linked to further options. This procedure follows a tree-like structure to get the desired outcome (Ahmadi, 2020).

The first step in the procedures of a decision tree regressor is fitting the regressor model using each input variable (Mohammed; Aung, 2016). The optimal split for each input variable is then chosen using the mean squared error, or other evaluation metric, with the maximum feature number for each split being set by the overall feature number. One significant drawback of using decision trees (DTs) is that

overfitting or trapping at the local minimum points can happen when a decision tree regression structure is very complex (Carrera; Kim, 2020) (Wang et al.; Li et al., 2018).

2.2.2 RANDOM FOREST

Adopting the ensemble learning of the bagging type with DT-based learners might result in identical constructed trees, indicating a lack of variation. To avoid this problem, employ RF which may be defined as modeling utilizing bagging and a random selection of predictors to assemble each node of the DT (Breiman, 2001). This procedure, together with selecting the optimal collection of features that describe the data, is a critical stage in this approach (Thakur; Kumar, 2018).

The bootstrap approach is used to create m samples from a training set consisting of n observations for the output variable and x predictors or input variables. Each sample is represented by a single regression tree, each node of which contains a randomly selected subset of the x predictor variables. After selecting the best subset, each regression tree generates an estimate of the response variable. The arithmetic average of these estimates is used to make a final prediction. This strategy aims to enhance regression tree performance by reducing variance (Assouline et al., 2018).

The estimated output $\hat{y}$, the number of trees m, and the predictions of *k*-th according to the input vector x are represented by the expression $f_k(x)$, which reflects the final prediction made by RF, where:

$$\hat{y} = \frac{1}{m}\sum_{k=1}^{m} f_k(x) \qquad (2.2.1)$$

2.2.3 LIGHT GRADIENT BOOSTING MACHINE

The boosting concept underpins the generalized boosted regression (GBM) model, which aims to minimize a loss function using an additive model (Friedman, 2001). To create models in the negative sense of the partial derivative of the loss function concerning the set of predictors, the GBM employs a gradient descent approach. The GBM is conducted iteratively to fit a regression model to the data and retrieve the residuals. First, a new prediction that incorporates the original forecast is

made, then a new model is adjusted for previous residuals, and lastly, a new residue measure is determined. Until a convergence requirement is satisfied, this procedure is repeated. While the GBM structure might enhance accuracy, it can also lead to overfitting. To avoid this, a regularization parameter for the learning rate is needed. Control hyperparameters employed in this dissertation for the GBM technique are the number of boosting iterations, maximum tree depth, shrinkage, and minimum terminal node size.

A large-scale library called LightGBM (Ke et al., 2017) performs gradient boosting and suggests several variations. The implementation of gradient boosting in this library aims to offer a computationally effective method that, similar to XGBoost, requires precomputation of the feature histogram. Furthermore, the library includes tens of learning hyper-parameters that allow this model to work in a variety of scenarios: on a GPU or a CPU, and it can do several randomizations, including bootstrap subsampling and column randomization, in addition to basic gradient boosting.

2.2.4 KALMAN FILTER NEURAL NETWORK

Artificial Neural Networks (ANNs) are computational models inspired by the human brain's structure and function. They consist of interconnected groups of artificial neurons, which process information by responding to inputs and transmitting outputs to other neurons. ANNs are used extensively in machine learning tasks due to their ability to learn complex patterns from data.

An ANN typically comprises three types of layers: the input layer, hidden layers, and the output layer. The input layer receives the initial data, while hidden layers are intermediate layers that perform computations and feature extraction. The output layer produces the final result. Each layer contains nodes, or neurons, which are the fundamental units of computation in a neural network. Neurons within a layer are connected to neurons in the subsequent layer through weighted connections. These weights are adjusted during the training process to minimize the error in predictions.

Neurons in an ANN process inputs by applying a linear transformation followed by a non-linear activation function. The output y of a neuron can be expressed as follows:

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{2.2.2}$$

x_i are the inputs, w_i are the weights, b is the bias, and f is the activation function. Several activation functions can be used in neurons, each introducing non-linearity into the model, enabling it to capture complex patterns.

Regression forecasting involves predicting a continuous output variable based on input variables. ANNs are well-suited for this task as they can model complex relationships between inputs and outputs. The steps for using ANNs in regression forecasting typically include data preprocessing, model design, training, and evaluation. By learning from historical data, ANNs can generalize and make accurate predictions on unseen data, making them powerful tools for regression forecasting in fields like finance, economics, and weather prediction.

Kalman Filter Artificial Neural Networks (KFNNs) integrate the principles of Kalman filtering with traditional neural network architectures. The Kalman filter is a recursive algorithm used to estimate the state of a dynamic system from noisy measurements. When combined with ANNs, it enhances the network's ability to handle time-series data and improves prediction accuracy in dynamic environments. In KFANNs, the Kalman filter can be used to update the weights of the ANN, providing a more robust and adaptive learning mechanism. This combination leverages the strength of ANNs in capturing non-linear patterns and the efficiency of Kalman filtering in processing temporal information, making it particularly effective in applications like tracking, navigation, and real-time forecasting.

2.3 EXPLAINABLE ARTIFICIAL INTELLIGENCE

Explainable Artificial Intelligence (XAI) is a branch of artificial intelligence that aims to make AI models more transparent and understandable through a set of techniques and methods developed for this purpose. The main goal is for everyone to be able to understand the decision-making process of an AI and effectively make the decision made by providing clear and accessible explanations while preserving performance and accuracy (Gunning et al., 2019).

The need for XAI techniques was based on the adoption of models such as Deep Neural Networks and other complex machine learning techniques. While these more complex models provide excellent performance, they fail to explain decisions made and internal operations due to a lack of transparency, as they are "black box" models. XAI, in addition to supporting a decision made by the model, also provides the importance/relevance of each variable for the output class.

There are various approaches and techniques used in the field of XAI such as Local Interpretable Explanations and Model Agnostic (LIME) (Ribeiro et al., 2016), SHapley Additive exPlanation (SHAP) (Lundberg et al., 2017), Gradient Class Activation Mapping (GRAD-CAM) (Selvaraju et al., 2019) and Importance of Attributes in Deep Learning (DeepLIFT) (Shrikumar et al., 2017).

Among the models mentioned, there are distinctions such as local, global and model-specific, and model-agnostic scopes. Lime is an example of a local scope technique because a new explanation map is generated at each point in the dataset. Shap, on the other hand, is a global scope technique where the aim is to provide an interpretation of the model as a whole. Even so, the two models mentioned are model-agnostic because they can explain any model, unlike the GRAD-CAM and DeepLIFT models which are model-specific.

Local explanations can help to understand why a specific decision was made, helping to increase the user's confidence in specific examples. Global explanations can help to understand the whole and can be used to optimize the model based on what has been learned. Intrinsic, or specific, methods can be useful if you need to train a new model, where understanding is key, but agnostic methods allow you to use already trained models or proven learning techniques.

2.3.1 SHAPLEY ADDITIVE EXPLANATIONS

Finding a consistent and impartial explanation of how each characteristic affects the model's prediction is commonly accomplished through the use of SHAP values.

Based on game theory, SHAP values provide each feature in a model with a numerical value representing its importance. Positive SHAP features have a positive influence on the prediction, whilst negative SHAP features have a negative effect. The effect's strength is indicated by its magnitude.

SHAP also has many useful properties that make it effective for interpreting models, such as:

- Additivity: This implies that each feature's contribution to the final prediction can be calculated separately and then combined. This characteristic makes it possible to compute SHAP values quickly, even for large datasets.
- Missingness: When characteristics are absent or unimportant for a prediction, SHAP values are zero. This ensures that irrelevant features do not affect the interpretation and makes SHAP values resistant to missing data.
- Local accuracy: The difference between the expected and actual outputs of the model for a given input is the sum of the SHAP values. In other words, for a given input, SHAP values offer a precise and local interpretation of the model's prediction.
- Consistency: When a feature's contribution varies, SHAP values remain unchanged despite model changes. Thus, even in cases when the model's design or parameters change, the SHAP values offer a consistent interpretation of the model's behavior.

2.4 CONFORMAL PREDICTION

Conformal prediction (CP) is a machine learning technique that aims to provide statistically reliable and calibrated predictions, with a measure of reliability associated with each prediction point assuming exchangeability of the data, other key features are:

- **Statistical Confidence Level Control:** The approach is based on statistical theory, ensuring that the error rate (confidence level) is controlled according to the value specified by the user.
- **Flexibility in Modeling:** It can be combined with different machine learning algorithms like statistical, machine learning, or deep learning methods, making it a flexible approach.
- **Robustness to Assumption Violations:** Conformal prediction is less sensitive to violations of model assumptions (drifting) than some classical techniques.

Traditional probabilistic forecasting approaches, such as linear regression and support vector machines (SVM), often also including specific methods for time series, provide measures of uncertainty in the form of confidence intervals. However, these measures are often based on simplified assumptions such as specific distribution or linear relationship and may not adequately capture the true uncertainty associated with forecasts, especially in time series contexts. Furthermore, classic approaches can be very sensible to outliers or drifting in the data. On the other hand, the CP technique aims to provide prediction intervals, point-to-point, that reflect the uncertainty inherent in the data, based on the principles of statistical reliability theory and probability.

CP uses the concept of a region of conformity. For a given confidence level there is a region of conformity $\Gamma^{1-\epsilon}$ where it contains the point with a level of at least $1-\epsilon\,\%$ of probability of belonging to that range. Therefore, generally the region of conformity $\Gamma^{1-\epsilon}$ contains the predicted point $\hat{y}$, which is usually the result of predictions in classical machine learning techniques. Numerically, if the confidence level is 95%, we have a region of conformity $\Gamma^{0.05}$.

The technique can be used both for regression and classification problems. For regression models, y is a number and $\Gamma^{1-\epsilon}$ a numerical interval around $\hat{y}$. As for the classification cases, y is a class and $\Gamma^{1-\epsilon}$ and a set of possible classes that $\hat{y}$ can take with that given probability level, where in the best case the result of $\Gamma^{1-\epsilon}$represents only one class.

Conformal prediction has different algorithms and methods available in the literature. One of the precursor algorithms is known as the inductive conformal prediction (ICP) algorithm by (Vovk et al., 2005) which belongs to the split conformal prediction group. Split-CP consists of dividing the train set into two other sets: train and calibration where all predictions will be fitted with the train set but all the conformity regions will be obtained by the adjustment process in the calibration set, so ICP is a sampling-based algorithm that generates prediction sets by randomizing the training data due to the property of exchangeability.

In addition to ICP, there are other variations of conformal prediction depending on the root problem, such as Classification (Binary, Multi-Class, or Multi-Label), Regression, or Calibration. Only the regression variations will be discussed in more detail later in this section like Naive, Split-CP, Jackknife and variations, Cross-validation and variations, conformized quantile regression (CQR), and ensemble batch prediction intervals (EnbPI) for time series.

Although conformal prediction has several advantages and is a valuable approach to dealing with statistical uncertainty, it also has some limitations and challenges. **1)** Can be computationally intensive, especially when the data set is large. Building conformity sets for each possible prediction can be computationally costly. **2)** Effectiveness depends on the performance of the underlying machine learning algorithm. If the base model is not suitable for the problem at hand, the performance of conformal prediction can be compromised. **3)** Requires careful calibration to ensure that the confidence intervals adequately match the specified error rate. Inadequate calibration can lead to excessively wide or narrow intervals. **4)** In situations where there is an imbalance in the classes of data, conformal prediction can produce confidence intervals that favor the majority class, making it less informative for the minority class. **5)** As the number of features (dimensions) increases, the effectiveness of conformal prediction can decrease due to the curse of dimensionality, where the data becomes more dispersed and building conformity becomes more challenging.

2.4.1 TRANSDUCTIVE CONFORMAL PREDICTION

Transductive conformal prediction (TCP) or full conformal prediction is the first method described by Gammerman, Saunders, Vapnik, and Vovk in the late 1990s. TCP is a theory-based method of generating valid prediction sets that makes no assumptions about the data distribution or the underlying prediction model and relies solely on exchangeability. That is, TCP provides finite-sample guarantees that a true value y is contained in a prediction set C with a probability of at least $1-\alpha$ for any level of incorrect coverage chosen by the user.

Given a set of exchangeable data pairs $\{(X_i, Y_i)_{i=1}^{n}\} \in \mathbb{R}^d \times \mathbb{R}$ and a chosen coverage level $1-\alpha$, TCP aims to generate a conformal set $C_n(x)$ for new data points x_{n+1}such that the true value lies on the set with probability $1-\alpha$.

Let A be the algorithm that maps a data set of any size to a fitted prediction function $\hat{\mu}$ such $A: \bigcup_{m\geq 1}^{(X\times Y)^m} \longrightarrow \{\hat{\mu}: X \to Y\}$. Algorithm A must also treat the data exchangeably, i.e. for any permutation π: $[m] \to [m]$ where $m = n+1$. If A employs randomness and its deficiency over all permutations are small, it suffices for the equality to hold in distribution as well.

$$\mathrm{A}(\{(x_1, y_1), \ \dots, (x_m, y_m)\}) = \mathrm{A}\left(\left\{\left(x_{\pi(1)}, y_{\pi(1)}\right), \ \dots, \left(x_{\pi(m)}, y_{\pi(m)}\right)\right\}\right) \qquad (2.4.1)$$

Then, given a set of covariates $x \in \{X_{n+1}, X_{n+2}, \dots\}$, equation 2.4.2 predicts $\{Y_{n+1}, Y_{n+2}, \dots\}$ training for all possible y values in the label space Y. The next step is choosing a nonconformity score function.

$$\widehat{\mu_y} = A(\{(X_1, Y_1), \dots, (X_n, Y_n), (x, y)\}) \qquad (2.4.2)$$

Defining $\{(X_1, Y_1), \dots, (X_n, Y_n)\}$ as a bag of old data and $\left(X_{n+1}, \widehat{Y_1}\right)$ the new data pair, nonconformity measure is a way to mathematically infer how different the new data is from old data. Formally, a nonconformity measure is a mapping to each possible bag of old data and each possible new data, demonstrated on equation

2.4.3, $\widehat{s_{\widehat{\mu_y}}}$ assigns a numerical score indicating how different the new data is from the old ones. Let's rename $\widehat{s_{\widehat{\mu_y}}}$ to $\widehat{s_y}$for simplicity from now on.

$$\widehat{s_y} : X \times Y \to \overline{\mathbb{R}} \tag{2.4.3}$$

There is an infinity of nonconformity score functions available, and it is easy to invent another one, but a common option is to use the absolute residuals, i.e. $\widehat{s_y} = \left|y - \widehat{\mu_y}(x)\right|$. Lastly, for every combination of (X_a, Y_b) onde, $a \in \mathbb{N} \leq n$ e $b \in \mathbb{N} \leq n$ every score $\widehat{s_y} < (1-\alpha)quantile$ of all scores in the original data pair $\{(X_i, Y_i)_{i=1}^{n}\}$, is chosen to construct the conformal set given in equation 2.4.4.

$$C_n(x) = \left\{y \in Y : \widehat{s_y}(x,y) \leq q_{\widehat{\left(1+\frac{1}{n}\right)(1-\alpha)}}\left(\{\widehat{s_y}(X_i, Y_i) : i \in [n]\}\right)\right\} \tag{2.4.4}$$

The guarantee that is got is that

$$\mathbb{P}\{Y_{n+1} \in C_n^{hat}(X_{n+1})\} \geq 1 - \alpha \tag{2.4.5}$$

Further on nonconformity measure subject, p-value is a technique also often used. The absolute difference between y_{true} and $\widehat{\mu_y}(x)$ may be scaled so the numerical value of $\widehat{s_y}$ does not say, by itself, how nonconformal (A) finds (x, y) to be. One solution: a comparison between μ_i and μ_j where $\mu_i = \widehat{\mu_y}$ shown in equation 2.4.6, in this case the lower number of p-value represent that (x, y) is very nonconforming (outlier), If it's larger, then (x, y) is very conforming.

$$\frac{\left|\{j=1,\ldots,n : \mu_j \geq \mu_i\}\right|}{n} \tag{2.4.6}$$

2.4.2 SPLIT METHOD

Split conformal prediction (Split CP), was initially developed by Papadopoulos et al. (2002); Vovk, Gammerman, and Shafer (2005) where is a particular case of full conformal prediction.

Unlike full conformal prediction, the split method does not need to recompute the conformal set for every new data point arrived and is much more computationally efficient once all combinations of (x_i, y_j) is not done. Split CP carries the concept of dividing the training dataset into two sets called train (D_{train}) and calibration (D_{cal}) such:

$$D_{train} \cup D_{cal} = \{1,2, \dots, n\}$$
$$D_{train} \cap D_{cal} = \emptyset$$

Summing up, Split CP will use the train set to fit the point predictor and the calibration set will be used on residual calculation $\widehat{s_y}$ before computing the conformal quantile $\widehat{q_{D_{cal}}}$. After all the conformal set is generated, group the predictor with D_{train} and the quantile with (D_{cal}). Nevertheless, the mathematical background of the split CP will be demonstrated.

Given a set of exchangeable data pairs $\{(X_i, Y_i)_{i=1}^{n}\} \in \mathbb{R}^d \times \mathbb{R}$ and a train and calibration set D_{train}, D_{cal}, respectively. Let A be the algorithm that maps the training data $\{(X_i, Y_i) : i \in D_{train}\}$ to a prediction function $\hat{\mu}$ where the model is trained one single time. Let $n_{train} + n_{cal} = n$ representing respective cardinalities.

$$A: (X \times Y)^{d_{train}} \longmapsto \hat{\mu}: X \to Y \tag{2.4.7}$$

$$\hat{\mu} = A(\{(X_i, Y_i) : i \in D_{train}\}) \tag{2.4.8}$$

The next step is evaluating the nonconformity score using the calibration set (D_{cal}) as shown in equation 2.4.9, define the conformal quantile on 2.4.10 and finally the conformal set on equation 2.4.11.

$$\widehat{s_y} = |y_i - \hat{\mu}(x_i)| \; i \in D_{cal} \tag{2.4.9}$$

$$\widehat{q_n} = \left\lceil \left(1 + \frac{1}{n_{cal}}\right)(1 - \alpha) \right\rceil \; smallest \; of \; \widehat{s_y}, \; i \in D_{cal} \tag{2.4.10}$$

$$\widehat{C_n}(x_{n+1}) = \widehat{\mu}(x_{n+1}) \pm \widehat{q_n}\{|Y_i - \widehat{\mu}(x_n)|\} \quad (2.4.11)$$

2.4.3 JACKKNIFE

Jackknife can be defined, nowadays, as a group of methods derivated from split conformal prediction. Leave-one-out (LOO) conformal prediction or Jackknife, was developed by Steinberger, L. and Leeb, H. (2016) and further enhanced by Barber et al. (2021) on jackknife+ and jackknife-minmax.

Given a set of exchangeable data pairs $\{(X_i, Y_i)_{i=1}^n\} \in \mathbb{R}^d \times \mathbb{R}$ and a train set D_{train}. Let A be the algorithm that maps the training data $\{(X_i, Y_i) : i \in D_{train}\}$ to a prediction function $\hat{\mu}_{-i}$ and $\hat{\mu}$.

The prediction point will be the same as split conformal prediction using $\hat{\mu}$ fitted to train set, but the interval calculation will differ once the calibration set does not exist in this method. Given the train set $\{D_{train}\}_{i=1}^n$, LOO takes one data pair $\{X_i, Y_i\}$ where i is the index of the removed point on a specific LOO model, resulting in n LOO models. LOO models are mathematically defined in equation 2.4.12, and the standard jackknife's conformal set is defined in equation 2.4.13 where the conformity score $\hat{s}_i^{LOO} = |Y_i - \hat{\mu}_{-i}(X_i)|$. Afterward, the jackknife+ is defined in equation 2.4.14 for better comprehension regarding the difference over the standard jackknife.

$$\hat{\mu}_{-i} = A\big((X_1, Y_1), \dots, (X_{i-1}, Y_{i-1}), (X_{i+1}, Y_{i+1}), \dots, (X_n, Y_n)\big) \quad (2.4.12)$$

$$\hat{C}_n^{jackknife}(X_{n+1}) = \hat{\mu}(X_{n+1}) \pm \widehat{q_n}\{\hat{s}_i^{LOO}\} \quad (2.4.13)$$

$$\hat{C}_n^{jackknife}(X_{n+1}) = \left[\hat{q}_n^{-}\{\hat{\mu}_{-i}(X_{n+1}) - \hat{s}_i^{LOO}\}, \hat{q}_n^{+}\{\hat{\mu}_{-i}(X_{n+1}) + \hat{s}_i^{LOO}\}\right] \quad (2.4.14)$$

$$\hat{C}_n^{jack-mm}(X_{n+1}) = \left[\min_{i=1,\dots,n} \hat{\mu}_{-i}(X_{n+1}) - \hat{q}_n^{+}\{\hat{s}_i^{LOO}\}, \max_{i=1,\dots,n} \hat{\mu}_{-i}(X_{n+1}) + \hat{q}_n^{+}\{\hat{s}_i^{LOO}\}\right] \quad (2.4.15)$$

Unlike split-CP, jackknife avoids overfitting issues during the training process but can lose predictive cover when $\hat{\mu}$ becomes unstable, when the sample size is close to the number of features, for example.

In contrast to standard Jackknife, Jackknife+ uses the LOO predictions $\hat{\mu}_{-i}(X_{n+1})$ for the test point, and the jackknife centers the interval on the predicted value $\hat{\mu}(X_{n+1})$, in other words, the conformal set on a standard jackknife is a symmetric and non-fixed interval around the prediction point for the test point and the jackknife+ is an interval, usually not symmetric and non-fixed, around the median prediction. Using the median prediction guarantees that the real value lies inside $\hat{C}_n^{jackknife+}(X_{n+1})$ for $\alpha \leq \frac{1}{2}$. Jackknife+ appears to be a slight modification of the standard version, but the changes guarantee to achieve predictive coverage at a level $1-2\alpha$. To correct this, jackknife-min-max uses a more conservative form of prediction interval step guaranteeing the target coverage $1-\alpha$ without any assumptions on the algorithm or distribution of the data.

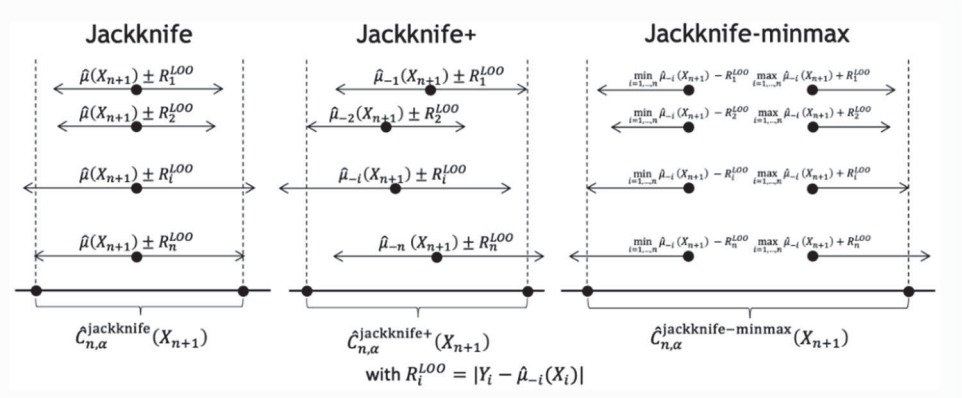


Figure 2.12 - Differences between Jackknife algorithms on interval steps handling
Source: Adapted from Barber et al. (2021)

2.4.4 CROSS-VALIDATION METHOD

Cross Conformal Prediction (CCP) was developed by Vovk, V. (2012) which is a hybrid of methods of Inductive Conformal Prediction (ICP) and cross-validation (CV) aiming to get a more predictively efficient model regarding the production of larger prediction sets.

ICP split the dataset into training and calibration sets where the training set will only be used by prediction function to point prediction of the data and the calibration set will only be used to calculate the conformity interval. On the other hand, Cross-CP

split the dataset into K disjoint folds where the full training set is used for calibration and most of the parts of the training set (80% - 90%) are used for prediction rules.

Given a set of exchangeable data pairs $\{(X_i, Y_i)_{i=1}^{n}\}$ and a train set D_{train} of size n. The train set must be split into K non-empty folds $Z_k,\ k = 1, \dots, K;\ K > 1$ known as a parameter of the algorithm and $(St_1, \dots, St_K)$ is an index set partition of $\{1, \dots, l\}$ to every k^{th} fold, as an example $l = \frac{n}{K}$ is a possible path. Let A be the algorithm that maps each fold $\{(X_i, Y_i) : i \in D_{train}\}$ to a prediction function $\hat{\mu}_{-Z_k}$ and $Z_{St_k} \to Z_i,\ i \in St_k$.

Compute the nonconformity score of observations in Z_{St_k} and (x, y) for each $k \in \{1, \dots, K\}$ and each possible label $y \in \mathbb{R}$ of the test point, respectively, as shown in equation 2.4.16 and equation 2.4.17. Also, the conformal set is shown in equation 2.4.18 which introduces randomizations into the method with $\tau \sim Unif[0,1]$ and the conformity score is $\hat{S}_i^{CV} = \left|Y_i - \hat{\mu}_{-St_{k(i)}}(X_i)\right|,\ i = 1, \dots, n.$

$$\hat{\mu}_{i,k} = A\left(Z_{St_{-k}}, Z_i\right) \tag{2.4.16}$$

$$\hat{\mu}_{k}^{y} = A\left(Z_{St_{-k}}, (x, y)\right) \tag{2.4.17}$$

$$\hat{C}_{n,K,\alpha}^{CV}(X_{n+1}) = \left\{y \in \mathbb{R} : \frac{\tau + \sum_{i=1}^{n} 1\left\{\left|y - \hat{\mu}_{-St_{k(i)}}(X_{n+1})\right| < S_i^{CV}\right\} + \tau 1\left\{\left|y - \hat{\mu}_{-St_{k(i)}}(X_{n+1})\right| = S_i^{CV}\right\}}{n+1} > \alpha\right\} \tag{2.4.18}$$

$$\hat{C}_{n,K,\alpha}^{CV+}(X_{n+1}) = \left[\hat{q}_{n,\alpha}^{-}\left\{\hat{\mu}_{-St_{k(i)}}(X_{n+1}) - S_i^{CV}\right\},\ \hat{q}_{n,\alpha}^{+}\left\{\hat{\mu}_{-St_{k(i)}}(X_{n+1}) + S_i^{CV}\right\}\right] \tag{2.4.19}$$

$$\hat{C}_{n,K,\alpha}^{CV-m}(X_{n+1}) = \left[\min_{i=1,\dots,n} \hat{\mu}_{-St_{k(i)}}(X_{n+1}) - \hat{q}_{n,\alpha}^{+}\{S_i^{CV}\},\ \max_{i=1,\dots,n} \hat{\mu}_{-St_{k(i)}}(X_{n+1}) + \hat{q}_{n,\alpha}^{+}\{S_i^{CV}\}\right] \tag{2.4.20}$$

Similarly to jackknife, CCP has variations such as CV+ and CV-minmax applied to the same purpose as jackknife method following the same coverage issue

where for any K the level will be $1-2\alpha$, particularly, since the excess noncoverage is at most $\sqrt{\frac{2}{n}}$, demontrated on Barber, R. (2020). The CV+ changes the symmetric characteristic of standard CV using the Leave-One-Out (LOO) technique on fold (not on data point as jackknife+) resulting on equation 2.4.19. As a matter of fact, the jackknife+ is a special case of the CV+ method when $K = n$. CV-minmax follows the same path as jackknife-minmax and is shown on equation 2.4.20. Both CV+ and CV-minmax has a tradeoff between model efficiency and intervals length, as an example with CV+, when a smaller K is chosen, will be computed only K rather than n models, but at cost of wider intervals since $\hat{\mu}_{-St_k}$ are fitted using lower sample size implying larger residuals.

2.4.5 CONFORMIZED QUANTILE REGRESSION (CQR)

Conformized Quantile Regression was initially developed by Romano et al. (2019) which is a segment of Split Conformal Prediction. As can be imagined, this technique is a hybrid of Split-CP and Quantile Regressions. Quantile regressors aim to estimate conditional quantiles instead of conditional mean of the response variable. The technique is more robust to outliers and has flexibility in estimating as many quantiles as needed between $[0,1]$ allowing to achieve a measure of tendency and dispersion.

Given a set of exchangeable data pairs $\{(X_i, Y_i)_{i=1}^{n}\} \in \mathbb{R}^d \times \mathbb{R}$ and a train and calibration set D_{train}, D_{cal}, respectively. Let A be the quantile algorithm that maps the training data $\{(X_i, Y_i) : i \in D_{train}\}$ to prediction function $\hat{\mu}^{\frac{\alpha}{2}}$, $\hat{\mu}^{1-\frac{\alpha}{2}}$, and let $n_{train} + n_{cal} = n$ representing respective cardinalities.

Given any quantile regression algorithm A and D_{train} set, two conditional quantile functions must be fitted, generating $\hat{\mu}^{\frac{\alpha}{2}}$, $\hat{\mu}^{1-\frac{\alpha}{2}}$ as shown in equation 2.4.21. Afterward, for each $i \in D_{cal}$, the evaluation of conformity score is demonstrated in equation 2.4.22 and the empirical quantile formula in equation 2.4.23. CQR can handle and adapt the interval lengths on heteroscedasticity data

$$\left\{\hat{\mu}^{\frac{\alpha}{2}},\ \hat{\mu}^{1-\frac{\alpha}{2}}\right\} = A(\{X_i, Y_i\}:\ i \in D_{train}) \tag{2.4.21}$$

$$\hat{s}_i = \max\left\{\hat{\mu}^{\frac{\alpha}{2}}(X_i) - Y_i,\ Y_i - \hat{\mu}^{1-\frac{\alpha}{2}}(X_i)\right\} \tag{2.4.22}$$

$$\hat{q}_{n,\alpha} = \lceil (1-\alpha)(n+1) \rceil\ smallest\ of\ \hat{s}_i,\ i \in D_{cal} \tag{2.4.23}$$

Finally, given a new data point X_{n+1}, the formula to compute the conformal set using CQR is shown in equation 2.4.24.

$$\hat{C}_n(X_{n+1}) = \left[\hat{\mu}^{\frac{\alpha}{2}}(x_{n+1}) - \hat{q}_{n,\alpha},\ \hat{\mu}^{1-\frac{\alpha}{2}}(x_{n+1}) + \hat{q}_{n,\alpha}\right] \tag{2.4.24}$$

2.4.6 ENSEMBLE BATCH PREDICTION INTERVALS (EnbPI)

Up to now, all conformal predictors have almost any requirements but data exchangeability, even CQR probably controls miscoverage under mild distributional assumption of exchangeability. The fact is that a time series does not meet the exchangeability criteria but the conformal prediction on these cases could be a huge enhancement on every application, so there must be a way to flexibilize even more a conformal prediction technique to handle time-dependent data, of course at a cost.

During past years, a lot of work has been invested into conformal predictors beyond exchangeable data, one of the popular approaches, by (Tibshirani et al., 2019), assumes a covariate shift, using weights into conformal predictors when the test data distribution and training data distribution are proportional. In this method, the weights are data-dependent, meaning that there must be a function of the data points (X_i, Y_i) to adjust for the known distributions shift, also, the covariate shift property must hold rather than exchangeability. (Cauchois et al., 2020) on top of the previous method, when the shifted test distribution lies in an f-divergence ball around the train data distribution. Nevertheless, both methods still assume i.i.d or exchangeable training data, meaning that they are not directly applicable to time series. Parallel to this, (Mao et al., 2020) developed a weighted conformal prediction based on spatial domain assigning high weights to data points that are closer to test point and lower, or zero,

weights to points far from test point, but the main issue is the distributional assumptions required to acquire the theoretical guarantees. On top of last research, (Barber et al., 2022) enhanced the method by allowing nonexchangeable data and nonsymmetric algorithms using ca overage gap, which is bounded by a weighted sum of total variation distances between residual vectors, extending the coverage level when is lower than itself under exchangeability. Also, the article proposes that weights are fixed to correct the coverage gap or assumes a distribution to decay the weights over samples. Meanwhile, Gibbs and Candès (2021) proposed an enhancement on sequentially adjust the significance level α during prediction by reweighting the significance level based on online coverage values on test set, reaching a near-valid coverage on sequential data. With this in mind, Xu and Xie, 2023, proposed a model-agnostic, distribution-free on data and provable asymptotic conditional coverage guarantees, without assuming data exchangeability, called Ensemble batch Prediction Interval (EnbPI) based on homogeneous ensemble learning (learners are generated with the same algorithm but different training data) with LOO predictions and residuals.

Given a model $f\colon \mathbb{R}^d \to \mathbb{R}$ where d is the dimension of the feature vector, EnbPI assumes a time series data generating process with the form of equation 2.4.25. The error process $\{\epsilon_t\}_{t\geq 1}$ is assumed to be stationary and strongly mixing, replacing the exchangeability requirement of conformal prediction.

$$Y_t = f(X_t) + \epsilon_t,\ t = 1,2 \ldots \tag{2.4.25}$$

Let $\{(x_t, y_t)\}_{t=1}^{T}$ to be the training set, so, given a prediction algorithm from the user, using the train set, let $\hat{f}_{-i}$ be a LOO estimator of f which is not trained in the i^{th} data (x_i, y_i). Let s be the prediction horizon, or batch size, or sliding window, such $s \geq 1$, and α be the significance level, the prediction interval will be $\left\{\hat{C}^{\alpha}_{T+i}\right\}_{i=1}^{s}$ for $\{Y_{T+i}\}_{i=1}^{s}$. Thus, the conformal set regarding the train set is shown in equation 2.4.26 where LOO prediction residual $\hat{\epsilon}_i$ and $\hat{\beta}$ are shown on equation 2.4.27 and equation 2.4.28, respectively.

$$\hat{C}_t^{\alpha} = \left[\hat{f}_{-t}(x_t) + \hat{\beta}\ quantile\ of\ \{\hat{\epsilon}_i\}_{i=t-1}^{t-T}, \hat{f}_{-t}(x_t) + \left(1 - \alpha + \hat{\beta}\right) quantile\ of\ \{\hat{\epsilon}_i\}_{i=t-1}^{t-T}\right] \tag{2.4.26}$$

$$\hat{\epsilon}_i = y_i - \hat{f}_{-i}(x_i) \tag{2.4.27}$$

$$\hat{\beta} = \underset{\beta \in [0,\alpha]}{\arg min} \left((1 - \alpha + \beta)\ quantile\ of\ \{\hat{\epsilon}_i\}_{i=t-1}^{t-T} - \beta\ quantile\ of\ \{\hat{\epsilon}_i\}_{i=t-1}^{t-T}\right) \tag{2.4.28}$$

Once ensemble learners are trained, it is used to predict the center of the interval which is constructed based on the quantile values of T residuals. The ensemble learners are assumed to model f well, but this assumption tends to fail when the window size s is large and long-term predictions are made, Jensen et al. (2022).

For this present work, regarding all the conformal prediction techniques presented, only EnbPI could support the need due to the non-exchangeable nature of the wind speed dataset, hence this method will be used with other decomposition techniques and different models to achieve a better result.

3 RELATED WORKS

Previous studies and articles in the field of wind speed time series forecasting or wind turbine power forecasting are generally divided into three categories: i) Statistical models, ii) Models based on machine learning, and iii) Hybrid models. They are further divided in terms of forecast horizon into Very Short Term, Short Term, Medium Term, and Long Term. Wind speed forecasting is generally used to enable and facilitate the forecasting of other variables or factors, as previously mentioned in the case of forecasting power generation in wind turbines.

There are divergences in terms of time windows for classifying the time series horizon; this work is based on the classes Very Short, Short, Medium, and Long Term (Liu et al., 2019), but a state-of-the-art reference may adopt another segmentation pattern. The time window for each horizon is shown in Figure 3.1.

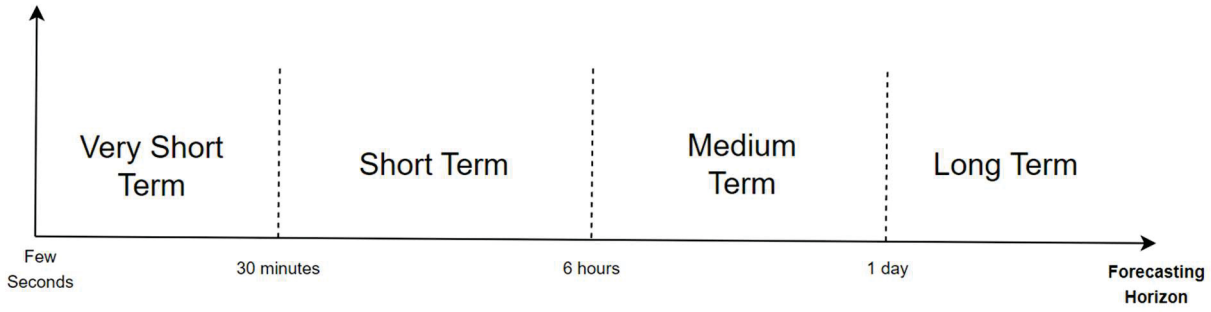


Figure 3.1 - Segmentation of Forecasting Horizons

Source: Adapted from Liu et al. 2019

The focus of Very Short-Term and Short-Term forecasts is to capture rapid changes in data for use in monitoring, operational safety, turbine control, network integration, stability, and resource reallocation (Nahid et al., 2020), (Mogos et al., 2022).

For these forecasting horizons, classical techniques can perform well, such as stochastic models like Autoregressive Moving Average (ARMA), Autoregressive and Integrative Moving Average (ARIMA), and Linear Forecasting. Other techniques used to improve very short-term and short-term forecasting are Artificial Intelligence

techniques, Physical Modeling, and hybrid methods such as combinations of neural networks or decomposition techniques with artificial neural networks.

Table 2 provides a summary of various studies on very short-term wind speed forecasting using different machine learning and statistical models across several regions. Riahy et al. (2018) employed Linear Prediction for WSF and evaluating performance with the Correlation Coefficient. In Thailand, Nahid et al. (2020) utilized CNN, LSTM, and CLSTM with performance measured by RMSE, MAPE, MAE, and Correlation. Mogos et al. (2022) in the USA applied MLP, Random Forest, KNN, and Decision Tree models for very short-term WSF. Wang et al. (2021) in China focused on using a combination of models including ARIMA, BP, GRNN, ENN, PSO-CE-ENN, WDO-CE-ENN, and MWS-CE-ENN, evaluated by MAE, MAPE, RMSE, and FE. These studies illustrate the diverse approaches and performance metrics applied in very short-term across different regions.

Table 2 - Summary of several very short-term wind speed forecasting models

Field	Techniques	Location	Metrics	Author
WSF	Linear Prediction	-	Correlation Coefficient	Riahy et al. (2018)
WSF	CNN LSTM CLSTM	Thailand	RMSE MAPE MAE Correlation	Nahid et al. (2020)
WSF	MLP Random Forest KFNN Decision Tree	USA		Mogos et al. (2022)
WSF	ARIMA BP GRNN ENN PSO-CE-ENN WDO-CE-ENN MWS-CE-ENN	China	MAE MAPE RMSE FE	Wang et al. (2021)

Table 3 presents a summary of some studies on short-term WSF using various machine learning and statistical models in different regions. Cadenas et al. (2010) conducted research in Mexico using ARIMA and ARIMA-ANN, evaluating the models with Mean Error (ME), Mean Squared Error (MSE), and Mean Absolute Error (MAE). In the USA, Liu et al. (2019) applied the SMNDE model, assessing performance with

metrics such as RMSE, R^2, Weight, ICPC, and CRPS. Zhu et al. (2012) utilized AR, Kalman Filter, RSTD, and TDD, with performance measured by MSE, MAE, MAPE, PCE, and MSAPE. Huang et al. (2011) focused on Penghu, Taiwan, using the GM model and evaluated the results with MAPE. Lastly, Jiang et al. (2017) researched in Peng Lai, China, employing Persistent, ARIMA, v-SVM-CS, G-ε-SVM-CS, G-v-SVM-PSO, and G-v-SVM-CS models, with performance metrics including MAE, MAPE, and RMSE. These studies highlight the diverse methodologies and evaluation metrics utilized in short-term WSF across different regions.

Medium-term forecasts have as their main objectives power system management, economic dispatch, load balancing, minor maintenance, and load shedding (Wang et al., 2021), (Liu et al., 2019), (Shukur et al., 2015).

Table 3 - Summary of several short-term wind speed forecasting models

Field	Techniques	Location	Metrics	Author
WSF	ARIMA ARIMA-ANN	Mexico	ME MSE MAE	Cadenas et al. (2010)
WSF	SMNDE	USA	RMSE R^2 Weight ICPC CRPS	Liu et al. (2019)
WSF	AR Kalman Filter RSTD TDD	-	MSE MAE MAPE PCE MSAPE	Zhu et al. (2012)
WSF	GM	Penghu - Taiwan	MAPE	Huang et al. (2011)
WSF	Persistent ARIMA v-SVM-CS G-ε-SVM-CS G-v-SVM-PSO G-v-SVM-CS	Peng Lai - China	MAE MAPE RMSE	Jiang et al. (2017)

Table 4 provides a summary of studies on medium-term wind speed forecasting using various machine learning and statistical models across different regions. Wang et al. (2015) focused on Xinjiang, China, utilizing ERNN, PERNN, PMERNN, and PAERNN models. The performance of these models was assessed using MAE, MSE, MAPE, and MPE. In Iraq and Malaysia, Shukur et al. (2015) conducted medium-term WSF using ARIMA, ANN, ARIMA-KF, and KF-ANN models, with MAPE as the evaluation metric. These studies highlight different approaches and performance metrics applied in medium-term WSF across various regions.

Table 4 - Summary of several medium-term wind speed forecast models

Field	Techniques	Location	Metrics	Author
WSF	ERNN PERNN PMERNN PAERNN	Xinjiang - China	MAE MSE MAPE MPE	Wang et al. (2015)
WSF	ARIMA ANN ARIMA-KF KF-ANN	Iraq and Malaysia	MAPE	Shukur et al. (2015)

The main objectives of Long-Term Forecasting are maintenance planning, operations management, turbine generator coupling or decoupling planning, operation cost optimization, generation capacity estimation (Devi et al., 2021), (Liu et al., 2019), (Barbounis et al., 2006).

Table 5 summarizes several significant studies on long-term forecasting using various ML and statistical models in different regions. Malik et al. (2016) conducted research in India focusing on long-term WSF using ANN, evaluating their model's performance with metrics such as MSE, R^2, and MAPE. Another study by Devi et al. (2021) in India explored a comprehensive range of methods, including ARIMA, VAR-TARCH, ARFIMA-APARCH, Markov Chain, ANN, KF-ANN, IIR-MLP, LSTM, and NARX, assessing their effectiveness using MAPE, MAE, and RMSE. In Crete, Greece, Barbounis et al. (2006) focused on long-term WPF and WSF employing models such as IIR-MLP, LAF-MLN, DRNN, static MLP, and FIR, with performance measured by MAE and RMSE. In Iran, Ahmadi et al. (2020) investigated long-term WPF using an array of ensemble methods, including Decision Tree, Bagging, Random Forest, AdaBoost, Gradient Boosting, and XGBoost, evaluating the models with MAE, RMSE,

and R^2. These studies collectively highlight the diverse methodologies and performance metrics utilized in long-term WSF and WPF across different regions.

Table 5 - Summary of several long-term wind speed forecasting models

Field	Techniques	Location	Metrics	Author
WSF	ANN	India	MSE R^2 MAPE	Malik et al. (2016)
WSF	ARIMA VAR-TARCH ARFIMA-APARCH Markov Chain ANN KF-ANN IIR-MLP LSTM NARX	India	MAPE MAE RMSE	Devi et al. (2021)
WPF/WSF	IIR-MLP LAF-MLN DRNN static MLP FIR	Crete - Greek	MAE RMSE	Barbounis et al. (2006)
WPF	Decision Tree Bagging Random Forest AdaBoost Gradient Boosting XGBoost	Iran	MAE RMSE R^2	Ahmadi et al. (2020)

In addition to the works already mentioned, there are also references where the forecast horizon is not fixed, i.e. the same technique is tested for different forecast horizons in order to analyze the adherence and accuracy of the model. Table 6 summarizes various studies on WSF and WPF using different techniques across multiple regions and forecast horizons. Silva et al. (2022) conducted research in Parazinho, Brazil, using VMD–SSA–STACK for very short to short-term WSF, evaluating their models with metrics such as IP, MAE, MAPE, RMSE, RRMSE, and SSE. Ambach et al. (2016) focused on short to medium-term WSF in Germany, employing techniques like AR, VAR, Lasso, Elastic Net, SVARX-TARCHX, and

ARFIMA–APARCH, with performance measured by RMSE and MAE. In Germany, Croonenbroeck et al. (2015) applied WPPT, GWPPT, Mycielski, AR, VAR, NP Regression, and Persistence for short to medium-term WPF, assessing the models with RMSE and MAE. Moreno et al. (2019) explored short to long-term WSF in Bahia, Brazil, using ARIMA, LSTM, VMD-SSA-ARIMA, and VMD-SSA-LSTM, with evaluation metrics including MSE, RMSE, MAE, MAPE, and RRMSE. Cai et al. (2019) applied AR + PCR + KF and SDA + SVR + UKF for short to long-term WSF, measuring performance with MAPE and RMSE. These studies highlight the diversity of approaches and performance metrics in WSF and WPF across different regions and forecast horizons.

Table 6 - Summary of several wind speed forecasting models for multiple horizons

Field	Techniques	Location	Metrics	Author
WSF	VMD–SSA–STACK	Parazinho - Brazil	IP MAE MAPE RMSE RRMSE SSE	Silva et al. (2022)
WSF	AR VAR Lasso Elastic Net SVARX-TARCHX ARFIMA–APARCH	Germany	RMSE MAE	Ambach et al. (2016)
WPF	WPPT GWPPT Mycielski AR VAR NP Regression Persistence	Germany	RMSE MAE	Croonenbroeck et al. (2015)
WSF	ARIMA LSTM VMD-SSA-ARIMA VMD-SSA-LSTM	Bahia - Brazil	MSE RMSE MAE MAPE RRMSE	Moreno et al. (2019)
WSF	AR + PCR + KF SDA + SVR + UKF	-	MAPE RMSE	Cai et al. (2019)

4 MATERIAL AND METHODS

This chapter presents more details about each case, statistics, and insights obtained during the forecasting process. The section 4.1 relates to a wind speed dataset gathered in Germany while the section 4.2 relates to a wind speed dataset gathered in Brazil. For both cases, the study intends to forecast from 1h to 15 days ahead.

4.1 BEUTENBURG

The German dataset from Jena used is from Weather Station of the Max-Planck-Institute for Biogeochemistry where the measure was made on top of the roof of the Institute Building. The documentation of the variables and the dataset is open source, but flattened variables can be seen in Table 7 with statistical information. The data is sampled every 10 minutes and has been updated every day since the deployment, so this study used the data from 2013 to 2021, as shown in Figure 4.1.

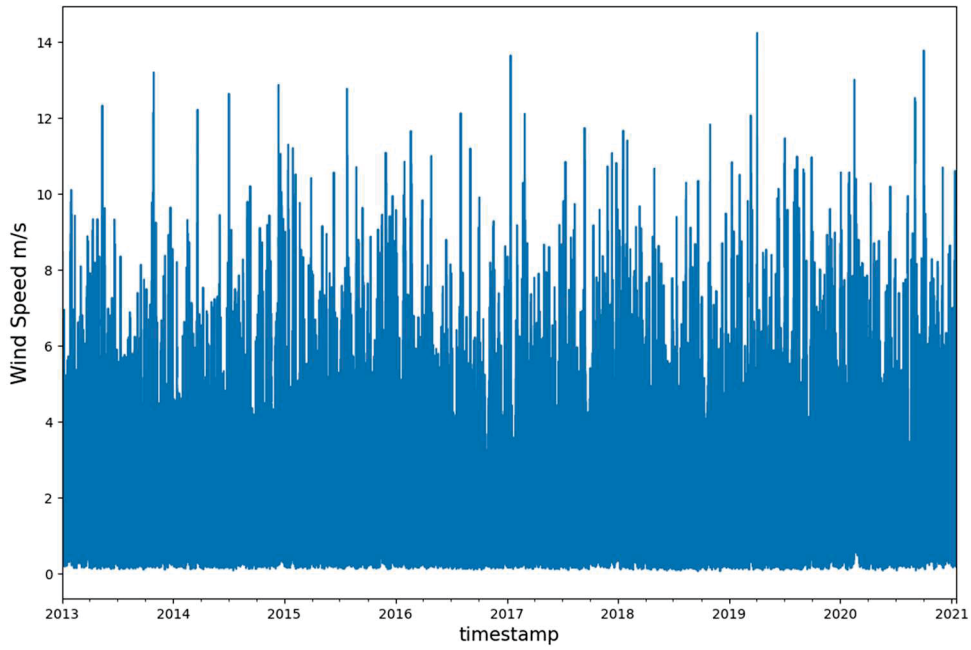


Figure 4.1 – Beutenburg dataset 2007 to 2018

Source: Author, 2024.

Table 7 displays the statistical metrics minimum (min), median, average, maximum (max), standard deviation (std), and percentile (%) for each time series used in this study. The datasets were divided into training and testing sets at 70% and 30%, respectively. The first 70% were utilized to train the adopted models, while the remaining 30% were used to evaluate the performance of the assessed forecasting models. In addition to giving the models additional information to learn about the

dynamics of wind speed forecasting, the adopted proportion of data used for training and testing the models also ensures that there is enough data to assess the performance of the suggested model.

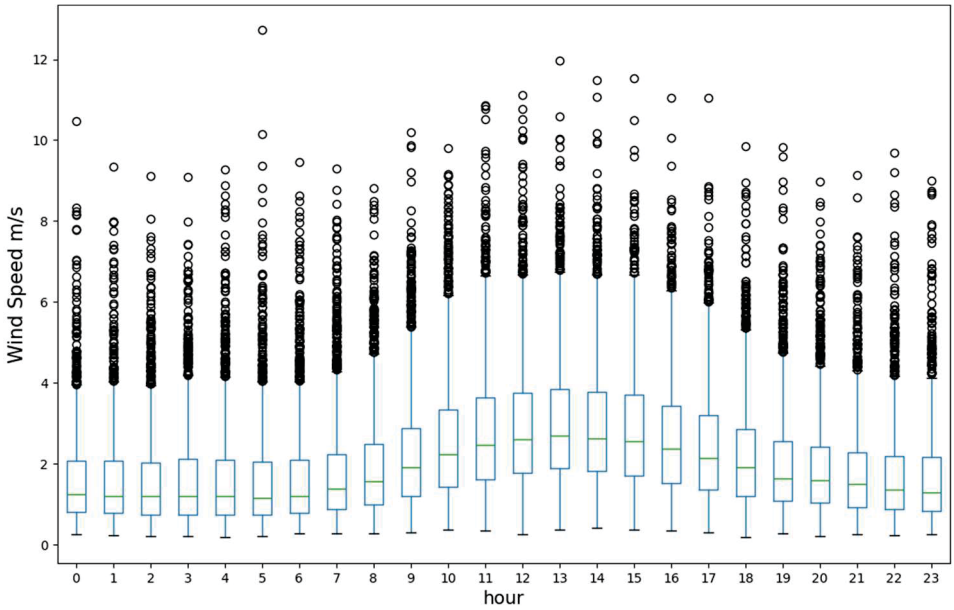


Figure 4.2 – Boxplot of hourly grouped wind speed of Beutenberg

Source: Author, 2024.

Features	mean	std	min	25%	50%	75%	max
p (mbar)	989.38	8.64	913.60	984.26	989.73	995.12	1020.07
T (degC)	9.68	8.23	-23.01	3.52	9.53	15.59	36.61
Tpot (K)	283.71	8.33	250.60	277.59	283.58	289.65	311.01
Tdew (degC)	5.05	6.53	-25.01	0.35	5.24	10.10	23.11
rh (%)	75.48	16.88	12.95	64.36	78.70	89.20	100.00
VPmax (mbar)	13.72	7.72	0.95	7.87	11.91	17.74	61.49
VPact (mbar)	9.55	4.11	0.79	6.27	8.88	12.38	28.32
VPdef (mbar)	4.17	5.02	0.00	0.92	2.25	5.44	47.17
sh (g/kg)	6.03	2.61	0.50	3.96	5.60	7.82	18.13
H2OC (mmol/mol)	9.66	4.16	0.80	6.35	8.98	12.51	28.82
rho (g/m**3)	1215.24	39.26	1059.45	1186.84	1213.55	1241.95	1393.54
wv (m/s)	2.16	1.57	0.00	1.00	1.79	2.92	16.83
max. wv (m/s)	3.60	2.40	0.00	1.77	3.00	4.84	27.88
wd (deg)	176.98	84.81	0.00	134.40	199.20	234.60	360.00
rain (mm)	0.01	0.11	0.00	0.00	0.00	0.00	18.90
raining (s)	49.78	154.17	0.00	0.00	0.00	0.00	600.00
SWDR (W/m^2)	123.83	207.37	0.00	0.00	2.41	165.39	1219.32
PAR ($\mu mol/m^2/s$)	242.82	402.76	0.00	0.00	7.92	329.73	2401.54
max. PAR ($\mu mol/m^2/s$)	297.43	496.45	0.00	0.00	11.20	390.93	2909.26
Tlog (degC)	20.00	8.21	-6.14	13.65	19.31	25.55	49.04
CO2 (ppm)	413.92	21.22	301.50	399.20	411.60	425.50	586.00

Table 7 - Summary of metrics of Case Beutenburg

4.2 LIMOEIRO

The gathered datasets correspond to wind turbine power generation sampled every 1 hour. All factors were tied to a wind turbine in a wind farm near Limoeiro, Pernambuco, Brazil. The measuring period ranges from 1980 to 2018, although for this investigation, the timeframe was 2007 to 2018 shown in Figure 4.3.

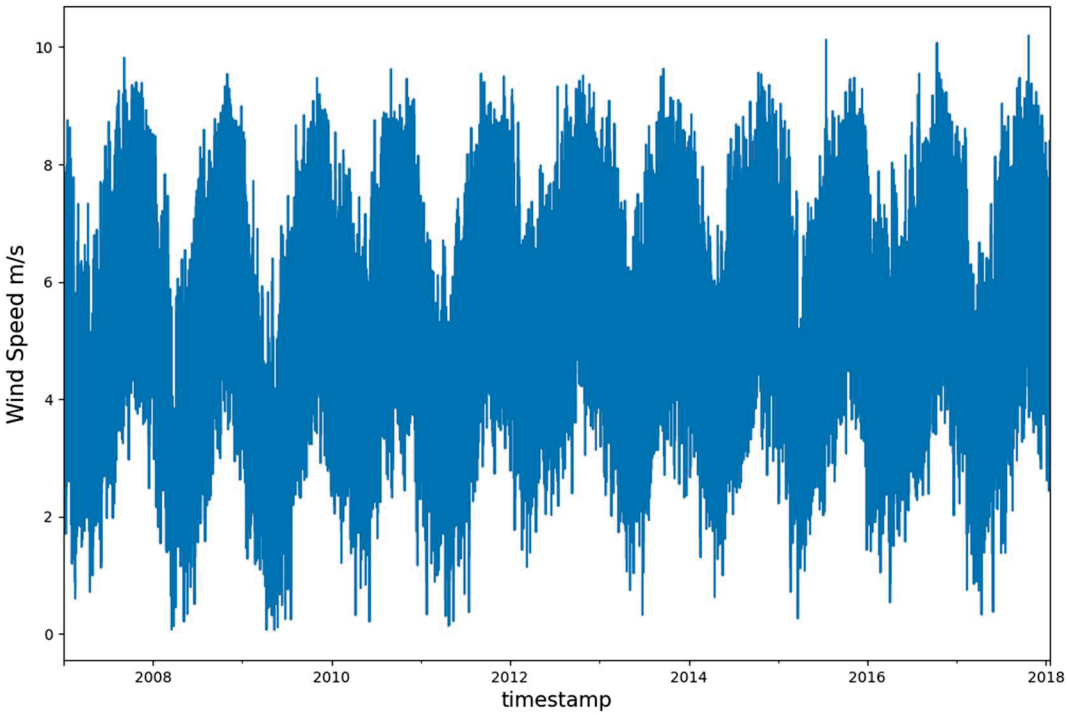


Figure 4.3 – Limoeiro Dataset 2007 to 2018

Source: Author, 2024.

Table 8 displays the statistical metrics minimum (min), median, average, maximum (max), standard deviation (std), and percentiles (%) for each time series used in this study. The datasets were divided into training and testing sets at 70% and 30%, respectively. The first 70% were utilized to train the adopted models, while the remaining 30% were used to evaluate the performance of the assessed forecasting models. In addition to giving the models additional information to learn about the dynamics of wind speed forecasting, the adopted proportion of data used for training and testing the models also ensures that there is enough data to assess the performance of the suggested model.

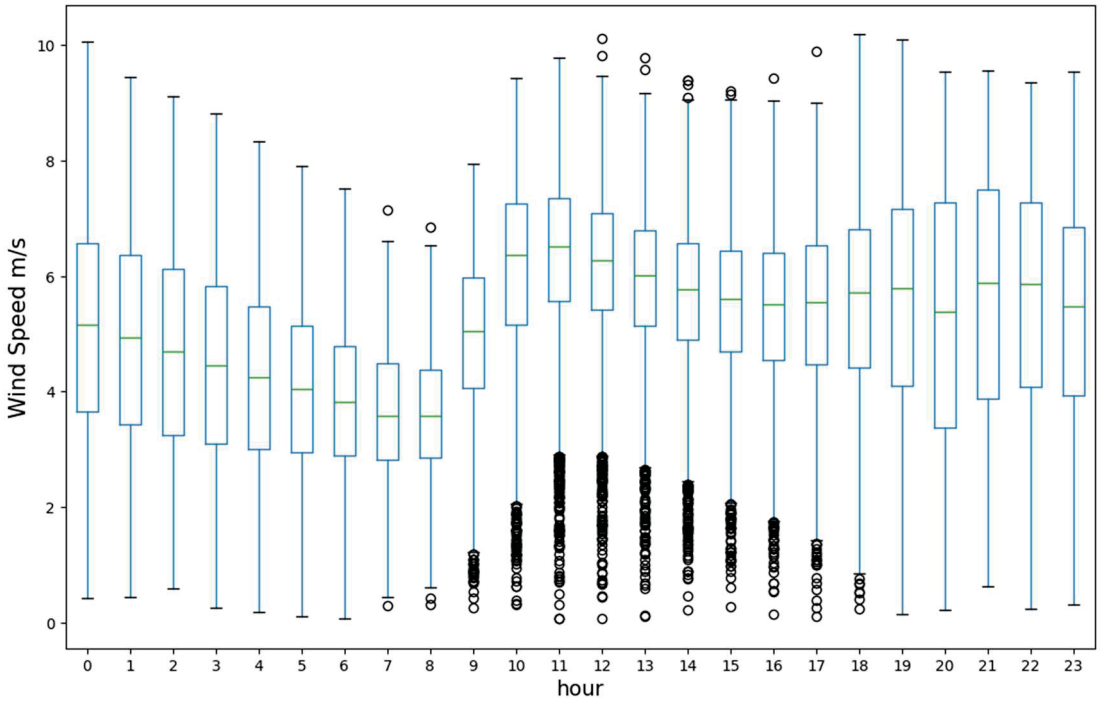


Figure 4.4 – Boxplot of hourly grouped wind speed of Limoeiro

Source: Author, 2024.

Features	mean	min	25%	50%	75%	max	std
Speed	5.12	0.06	3.73	5.17	6.48	10.47	1.77
Hour	11.50	0.00	5.75	11.50	17.25	23.00	6.92
Day	15.73	1.00	8.00	16.00	23.00	31.00	8.80
Month	6.52	1.00	4.00	7.00	10.00	12.00	3.45
Year	1999.00	1980.00	1989.00	1999.00	2009.00	2018.00	11.25
Wsdir	181.69	0.04	158.98	180.19	205.86	359.93	30.86

Table 8 - Summary of metrics of Case Limoeiro

Source: Author, 2024.

5 RESULTS AND DISCUSSION

In this chapter are presented the results and discussion for each case of study. For every case of study will be used the same workflow to achieve the best model where measures of uncertainty associated with the data will be used, with a selection of features based on the importance of the variables and optimization of the hyperparameters, presented in Figure 5.1.

Given a dataset, the first block will handle the initial visualization of the data, searching for visual discrepancies, outliers, patterns, and other insights.

The second block, Explanatory Data Analysis, aims to look deeply into statistics metrics of each dataset variable such as minimum, maximum, p-value, distribution, trend, and seasonality to address this information to the Preprocessing block that will, in fact, make necessary changes into the dataset such outliers removal, transformations or normalization, fulfillments into variables in case of missing values, split into train and test set, and so on.

The fourth block only objective is to split the target variable of the train set, into 2 or more variables regarding its seasonality, trend, noise, and other intrinsic harmonic components, in this block the VMD, SSA, X-11, or STLD will be applied.

The fifth block, Data Forecasting using ML, handles the forecasting itself of the target variable on train set, this step the Random Forest, Decision Tree, LightGBM, and KFNN will be applied and delivered to the sixth block, the Forecasting Metrics Analysis.

The sixth block aims to only acquire the performance of each model and delivers to the seventh block, Hyperparameters Optimization, which will seek the best set of hyperparameters for that specific model using Bayesian optimization and a limit of 300 iterations per model.

The loop of blocks 5, 6 and 7 ends after 300 iterations and only the best hyperparameters found will proceed to the eighth block, Conformal prediction, which aims to use the base model previously trained to make a probabilistic forecast using conformal prediction. In this block, the Enbpi will be applied once the problem is a time series forecasting.

The ninth block, Feature importance validation, and the tenth block, Feature selection, will seek the feature importance of each variable for each model and display it in order to check possible cuts into features that will not or poorly affect the model

accuracy and increase the model complexity and processing time. After this analysis the decision to change, the model should return to the fifth block.

Once all is done the model acquired is ready to go to production or any Machine learning operations (MLOps) Cycle. For comparison blocks 5 and 6 also generate outputs in this study, where the 5th block generates outputs using or not decompositions provided in the previous step and the 6th block generates output using conformal prediction with or without decompositions.

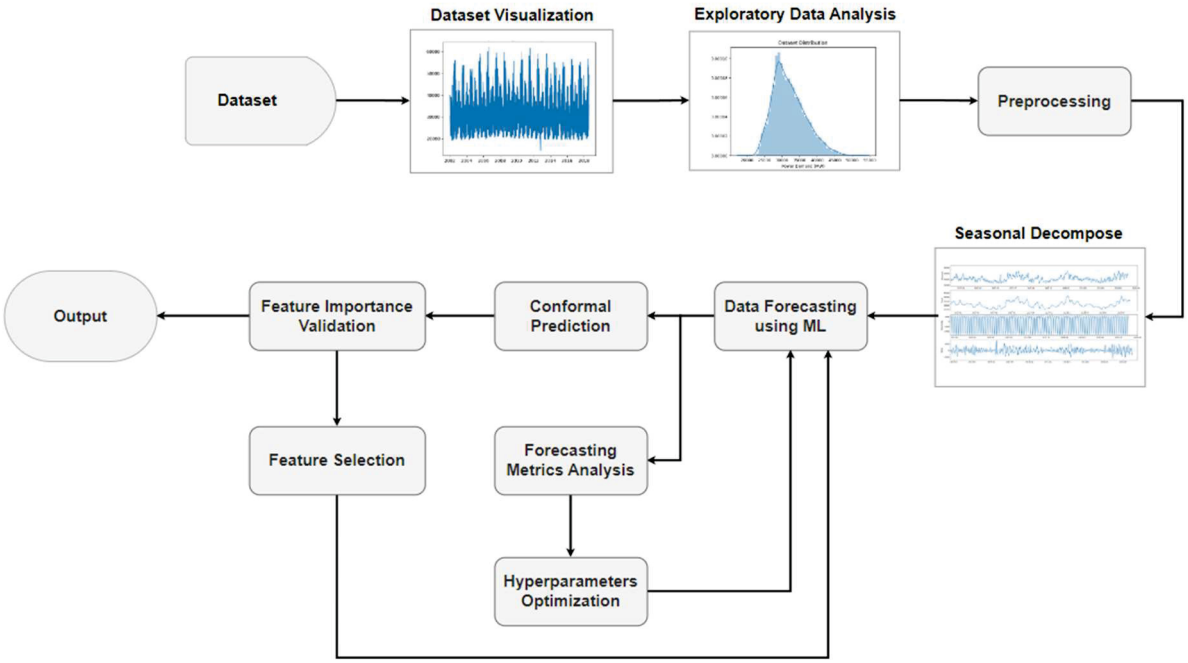


Figure 5.1 - Workflow of Wind Speed Forecasting

Source: Author, 2024

5.1 CASE BEUTENBURG

The performance metrics attained by every forecasting model under evaluation are listed in Table 9. Furthermore, the test and training outcomes are ranked according to lower MAPE. Lastly, the models that showed the lowest metrics in the test set are the best ones.

In general, the approaches based on the ensemble approach deliver good performance for short-term forecasting of wind speed about other models utilized, according to an evaluation of the test set findings shown in Table 9.

Train Set			Test Set			
RMSE	SMAPE	R^2	RMSE	SMAPE	R^2	Technique
0.18285	8.03238	0.98572	**0.25031**	**10.57023**	**0.97392**	**LGBM+VMD+SSA**
0.09583	**3.66565**	**0.99608**	0.25399	10.71841	0.97315	RF+VMD+SSA
0.29944	12.88314	0.96383	0.30409	12.63978	0.96150	LGBM
0.31164	13.25336	0.96082	0.31384	12.82736	0.95900	RF
0.50382	20.08356	0.80398	0.38610	14.63150	0.93794	KFNN
0.45266	18.04433	0.85652	0.50560	21.70502	0.89358	KFNN+VMD+SSA
0.59514	23.72405	0.84876	0.58386	27.74327	0.85809	DT+VMD+SSA
0.67070	28.97633	0.81855	0.67341	28.17854	0.81122	DT

Table 9 - Results for the Beutenburg case study

Source: Author, 2024

The increase of model robustness using the decomposition of the dataset with VMD and SSA helps to gain more understanding of the signal and its low to high order frequency but also increases the complexity of the model and processing time.

LGBM outperformed other techniques in this study followed by Random Forest, especially with the introduction of more decompositions of the signal into the model as exogenous variables. For instance, the input signal was split into 50 different signals on the SSA step where 16 of them were understood as proper signals and the rest of them as high-order frequency signals with minor interference on the main signal and noise. For the VMD decomposition, the reconstructed signal of SSA was used and split once again into 7 different signals, where are fed into the model's exogenous variables. However, any of these are probabilistic forecasting, meaning that the point prediction can vary in an unknown set. In production, the model should be able to deliver the point prediction and a statistically proven set where the actual y_{t+1} will be. For this, all the models above were fed into conformal prediction where the result of each model is shown in Table 10. To achieve this, an alpha of 0.05 was used, translating to a coverage level of 95%, with EnbPi model (once is a time series forecasting), cross-validation with overlapping blocks of length equal to 24, 50 resampling blocks and aggregation function as mean of the blocks.

Conformal prediction is having a correction of prediction set using partial fit or the standard way where the set is calculated at the beginning of the operation. In both cases, the application should determine which metric is more important, coverage or

width, once it's a trade-off graphic. Regarding wind speed forecasting the width of the CP model will decide the battle between these two metrics leading VMD and SSA models as the winners, especially the LGBM, which also was the best one before. Conformal prediction does not affect the accuracy of the model and gives the statistic trust set of predictions. For KFNN the conformal prediction needs an implementation of fit and iterative fit that is not implemented in the library used.

Dataset	Partial Fit		No Partial Fit			
	Coverage	Width	Coverage	Width	RMSE	Technique
Beutenberg	0.9420	**0.9770**	0.9440	**1.0080**	0.25031	**LGBM+VMD+SSA**
	0.9430	1.0180	0.9480	1.0620	0.25399	RF+VMD+SSA
	0.9510	1.2750	**0.9510**	1.2750	0.30409	**LGBM**
	0.9500	1.3040	0.9500	1.3030	0.31384	RF
	0.9200	1.7420	0.9260	1.8130	0.58386	DT+VMD+SSA
	0.9420	2.5430	0.9420	2.5430	0.67341	DT
	-	-	-	-	0.38610	KFNN
	-	-	-	-	0.50560	KFNN+VMD+SSA

Table 10 - Conformal Prediction Results Beutenburg

Source: Author, 2024

Analyzing the feature importance of each winning model, it was found that the models had a similar feature importance that is shown in Figure 5.2. It shows the 15 more important variables to the LGBM model and the difference between LGBM and RF was the "Lag_1" and some orders of VMD variables. Regarding the feature importance graphic, it is shown that the contribution of the SSA feature is greater when the SSA values increase; on the other hand, the lower the SSA values, the greater the negative contribution to the model, but still not as great as the positive contribution of the feature. The same principle occurs with max. speed feature as the SSA feature but it's simple to understand since the greater the max wind speed, the greater will be the mean wind speed in that time window. Nevertheless, the Lag_2 feature shows a small contribution on weight to the model, but the greater the feature value, the greater is the negative contribution to the model and vice versa, meaning that the greater the lag 2 of the wind speed variable, the lower the actual wind speed value tends to be. The fourth and last one that will discuss in this case, the wind direction goes back and forth over the positive-negative contribution but it's easy to relate that the reason is the

circumference that goes from 0 to 360 degrees. The conformal prediction result using LGBM+VMD+SSA with partial fit, or residual updates, can be seen in Figure 5.3 where the coverage was 0.942 and the width was 0.977.

Model	Parameters					
	max_depth	learning_rate	n_estimators	colsample_bytree	subsample	reg_alpha
LGBM+VMD+SSA	44	0.02514	962	0.73154	0.33078	0.94373
RF+VMD+SSA	15		903			

Table 11 - Hyperparameters of case Beutenburg (winner models)

Finally, on Table 11 are presented the hyperparameters of the models that this work found interesting in wind speed forecasting, including the winner and other models very close to the winning model.

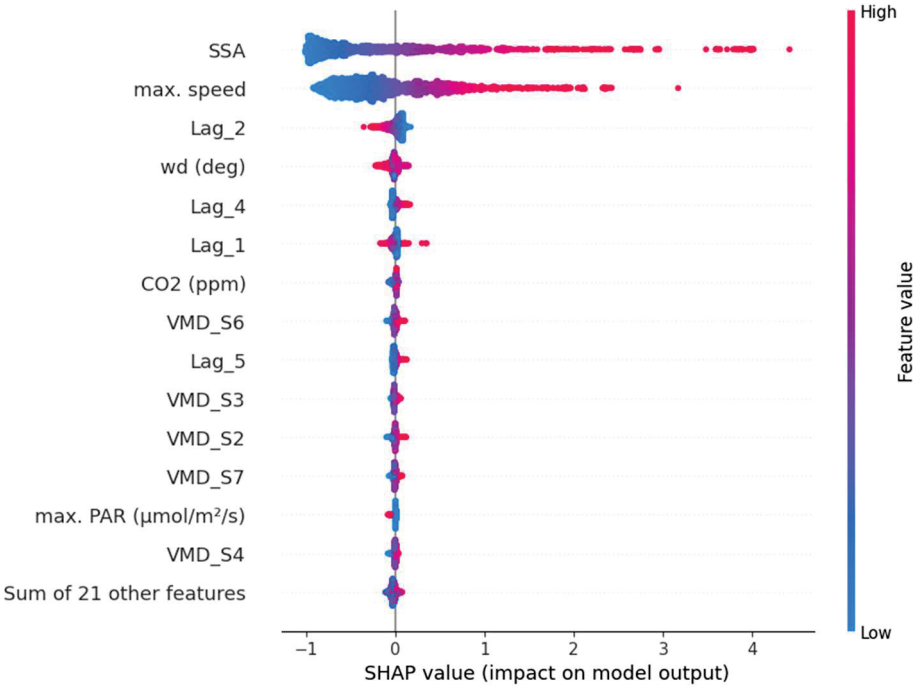


Figure 5.2 - Feature Importance LGBM+VMD+SSA Beutenburg

Source: Author, 2024

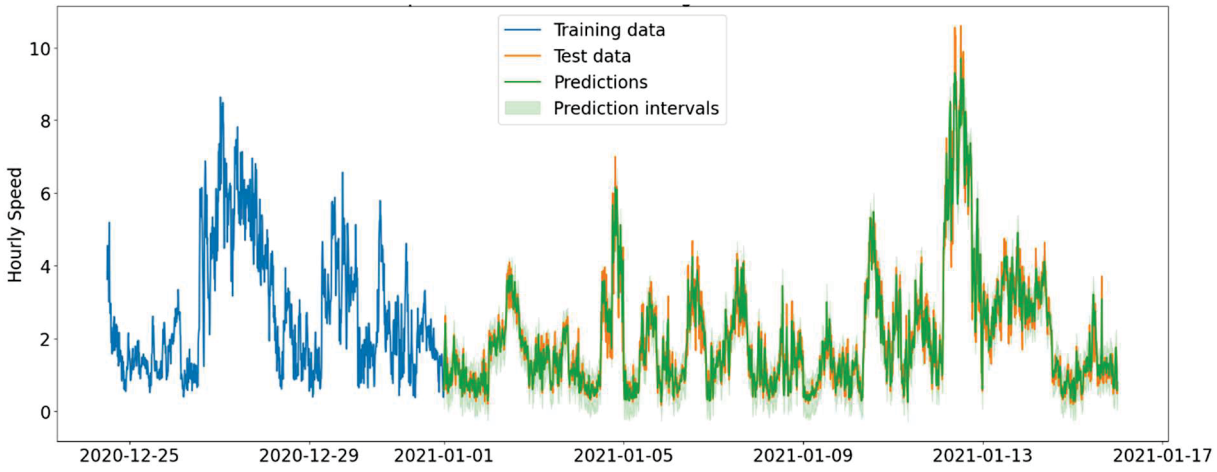


Figure 5.3 - Conformal Prediction LGBM+VMD+SSA Beutenberg with residual update

Source: Author, 2024

5.2 CASE LIMOEIRO

The performance metrics attained by every forecasting model under evaluation are listed in Table 12. Furthermore, the test and training outcomes are ranked according to lower sMAPE. Lastly, the models that showed the lowest metrics in the test set are the best ones.

Dataset	Train Set			Test Set			
	RMSE	SMAPE	R^2	RMSE	SMAPE	R^2	Technique
Limoeiro	0.17327	2.89512	0.99024	**0.18104**	**2.66080**	**0.98226**	LGBM
	0.08472	1.31524	0.99767	0.19127	2.80543	0.98020	RF
	0.21669	2.80856	0.91112	0.21360	3.30210	0.97531	KFNN+VMD+SSA
	0.07439	0.90850	0.99663	0.21597	3.40964	0.97475	LGBM+VMD+SSA
	0.07051	**0.87040**	**0.99697**	0.22548	3.57986	0.97248	RF+VMD+SSA
	0.44895	5.81883	0.87713	0.65417	10.82057	0.76839	DT+VMD+SSA
	0.74619	13.05248	0.81902	0.69083	11.23162	0.74170	DT

Table 12 - Results for the Limoeiro case study

Source: Author, 2024

An analysis of the test set data shown in Table 12 indicates that, in this case, the ensemble approach-based techniques perform worse for short-term wind speed forecasting than other models used. The use of VMD and SSA to decompose the dataset should improve model resilience and comprehension of the signal at low to high order frequencies, but it also increases model complexity and processing time. In this case, the performance was not as great as shallow models.

LGBM outperformed the other approaches in this investigation, followed by Random Forest. In the SSA stage, the input signal was divided into 70 separate signals, 14 of them were understood as proper signals and the rest of them as high-order frequency signals with minor interference on the main signal and noise. For the VMD decomposition, the SSA reconstructed signal was separated into 3 independent signals, which were then supplied into the model's exogenous variables. Although any of these are probabilistic forecasting, meaning that the point prediction can vary in a set that is unknown. In production, the model should be able to deliver the point prediction and a statistically proved set where the actual y_{t+1} will be. For this, all the models above were fed into conformal prediction, where the result of each model is shown on Table 13. To achieve this, an alpha of 0.05 was used, translating to coverage level of 95%, with EnbPi model (once is a time series forecasting), cross validation with overlapping blocks of length equal to 24, 50 resampling blocks and aggregation function as mean of the blocks.

Dataset	Partial Fit		No Partial Fit			
	Coverage	Width	Coverage	Width	RMSE	Technique
Limoeiro	0.91100	**0.67600**	0.90300	**0.67800**	0.21597	**LGBM+VMD+SSA**
	0.91700	0.81300	**0.92500**	0.79700	0.22548	**RF+VMD+SSA**
	0.78900	1.05500	0.66900	0.93700	0.18104	LGBM
	0.86900	0.96800	0.84700	0.95100	0.19127	RF
	0.85000	1.67200	0.82800	1.61400	0.65417	DT+VMD+SSA
	0.82200	3.25800	0.79400	3.11600	0.69083	DT
	-	-	-	-	-	KFNN+VMD+SSA

Table 13 - Conformal Prediction Results Limoeiro

Source: Author, 2024

Conformal prediction involves correction of the prediction set using partial fit or the usual method, in which the set is calculated at the start of the procedure. When faced with a trade-off visual, the application should decide which statistic is more important: coverage or width. Regarding wind speed forecasts, the breadth of the CP model will decide the battle between these two measurements, with the VMD and SSA models emerging victorious, particularly the LGBM. In the last result the regular LGBM was the best and in this case, this is also a good solution, if and only if, the coverage

and width satisfies the application. Conformal prediction has no effect on the model's accuracy and provides a set of predictions that the user may trust. Conformal prediction for KFNN requires fit and iterative fit implementations, which are not provided by the library.

Analyzing the feature importance of each winning model, it was found that the models had a similar feature importance that is shown on Figure 5.4. It's shown the 15 more important variables to the LGBM model and the difference between LGBM and RF was some orders of VMD variables. Regarding the feature importance graphic, it is shown that the contribution of the SSA feature tends to be lower, the greater the negative contribution to the model, on the other hand, the greater the feature value, the greater the positive contribution tends to be, but still not as great as the negative contribution of the feature. The same principle occurs with Lag_1 feature as the SSA feature. Nevertheless, lag_2 feature shows a small contribution on weight to the model, but the greater the feature value, the greater is the negative contribution to the model and vice versa, meaning that the greater the lag 2 of the wind speed variable, lower the actual wind speed value tends to be. The conformal prediction result using LGBM+VMD+SSA with partial fit, or residual updates, can be seen in Figure 5.5 where the coverage was 0.911 and the width was 0.676. Finally, on Table 14 are presented the hyperparameters of the models that this work found interesting on wind speed forecasting, including the winner and other models very close to the winning model.

Model	Parameters					
	max_depth	learning_rate	n_estimators	colsample bytree	subsample	reg_alpha
LGBM+VMD+SSA	44	0.02514	962	0.73154	0.33078	0.94373
RF+VMD+SSA	15		903			
LGBM	78	0.08526	980	0.93634	0.77053	2.05657
RF	49		998			

Table 14 - Hyperparameters of case Limoeiro (winner models)

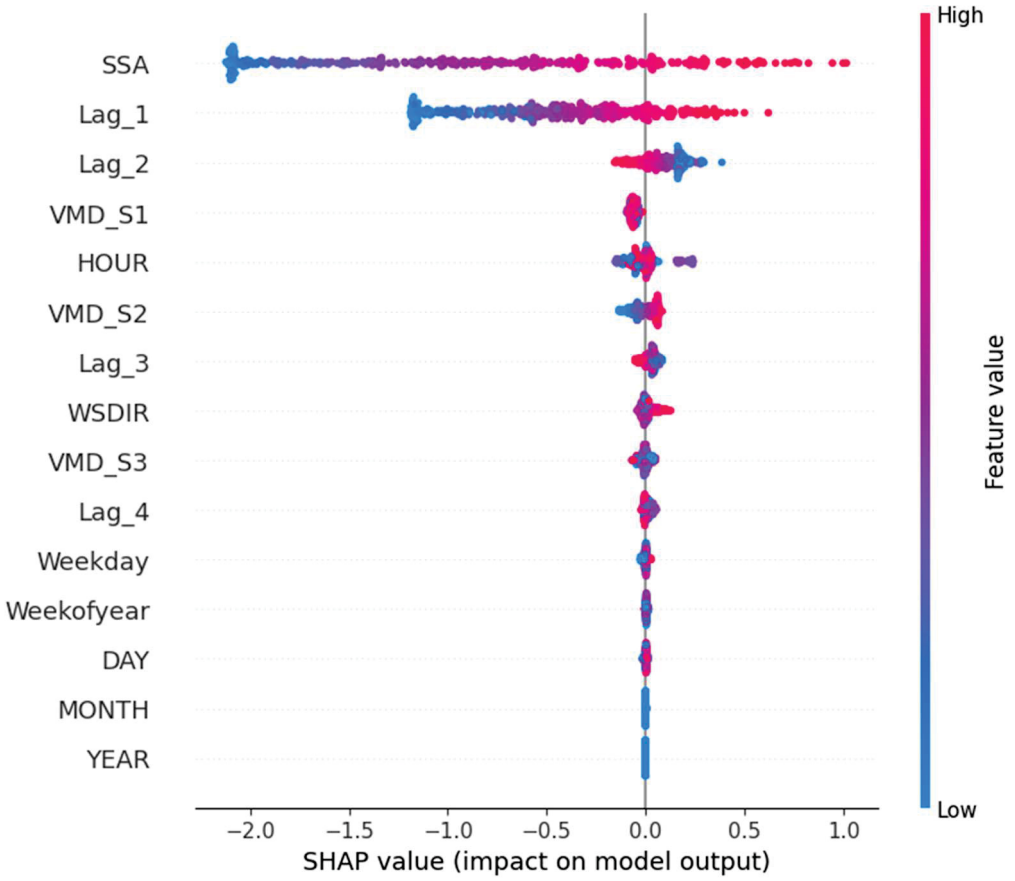


Figure 5.4 - Feature Importance LGBM+VMD+SSA Limoeiro

Source: Author, 2024

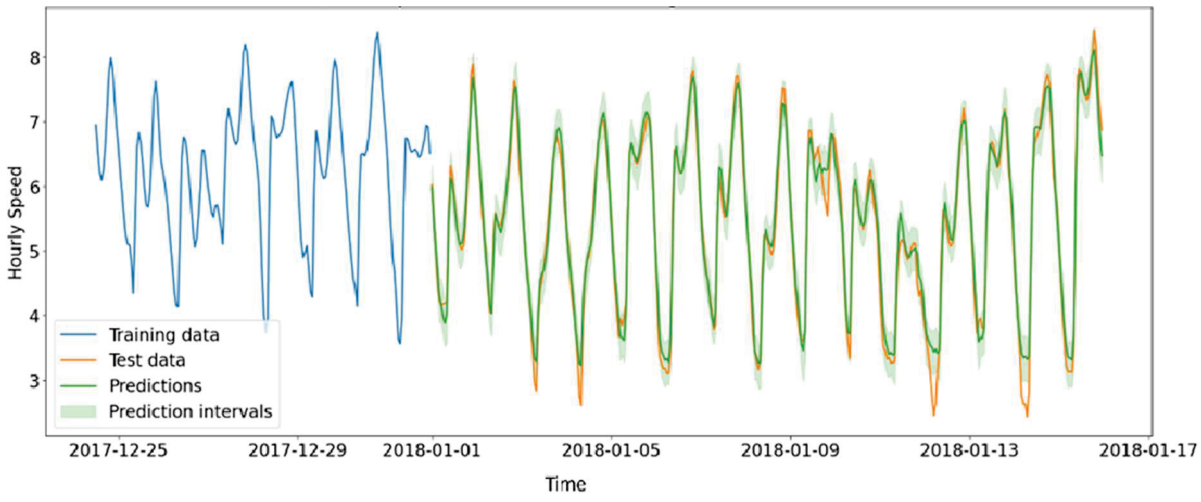


Figure 5.5 - Conformal Prediction LGBM+VMD+SSA Limoeiro with residual update

Source: Author, 2024

6 CONCLUSION AND FUTURE WORKS

This case study aimed to compare the predictive performance of the LGBM, DT, RF, and KFNN models and variations using the SSA and VMD decompositions trying to take off the noise and return the new variables as exogenous ones to the model. Following that, conformal prediction was used to feed these models in order to provide probabilistic forecasting for each model and its variants. For short-term forecasting, two wind speed forecasting datasets were used: dataset 1, the Beutenberg from Germany, and dataset 2, the Limoeiro from Brazil. Consequently, with the models and the results, the major objective was achieved successfully during the study, especially regarding probabilistic forecasting of short-term wind speed forecasting.

This work is intended, also, to present different studies in the field of wind speed forecasting as a target variable or as a source variable, as in the case of wind power forecasting in multiple time series horizons, from very short-term to long-term forecasting. During the work, many different time series decompositions and/or hybrid ML models were considered and studied and the main one was chosen to test the conformal prediction applied to those techniques.

The hyperparameter optimization used was a Bayesian optimization with cross-validation searching to minimize the negative average cross-validation score with 5 folds, for this purpose, the Optuna framework was applied with RMSE, sMAPE, and R^2. Furthermore, this work is intended to apply XAI methods for feature importance and support feature selection, specifically using Shapley.

The information contained in this study is relevant for decision-making aid and synthesizing the main factors that affect wind power generation, wind farms, consumers, and other sensitive outdoor applications. Based on the results presented here, small to large producers and managers of cooperatives are supported by concrete information that allows them to choose the main electricity generator at each time in the future to deliver the proper amount of electricity to the consumers or the network. This study provides proper information to extend the case to a longer forecasting period to help investors and companies plan the best area with the maximum generation during the year. All forecasting done in this study can be used to

feed other machine learning models to predict the power for each turbine in a wind farm, or a wind farm as a whole. This study is also sensitive to drastic climate changes that could cause a drift in the models. This aspect of the study should be enhanced in the near future. Even with positive outcomes, though, one must exercise caution when making decisions since wind is hard to predict and tends to fluctuate greatly over time, especially while working with bigger time horizons.

REFERENCES

Nahid, F. A., Ongsakul, W., and Manjiparambil, N. M. (2020). Very short-term wind speed forecasting using convolutional long short term memory recurrent neural network. *Proceedings of the 2020 International Conference and Utility Exhibition on Energy, Environment and Climate Change, ICUE 2020*. doi:10.1109/ICUE49301.2020.9307061

Ahmadi, A., Nabipour, M., Mohammadi-Ivatloo, B., Amani, A. M., Rho, S., and Piran, M. J. (2020). Long-term wind power forecasting using tree-based learning algorithms. *IEEE Access*, *8*. doi:10.1109/ACCESS.2020.3017442

Ambach, D., and Croonenbroeck, C. (2016). Space-time short- to medium-term wind speed forecasting. *Statistical Methods and Applications*, *25*. doi:10.1007/s10260-015-0343-6

Assouline, D., Mohajeri, N., and Scartezzini, J. L. (2018). Large-scale rooftop solar photovoltaic technical potential estimation using Random Forests. *Applied Energy*, *217*. doi:10.1016/j.apenergy.2018.02.118

Barber, R. F., Candès, E. J., Ramdas, A., and Tibshirani, R. J. (2021). Predictive inference with the jackknife+. *Annals of Statistics*, *49*. doi:10.1214/20-AOS1965

Barber, R. F., Candès, E. J., Ramdas, A., and Tibshirani, R. J. (2023). Conformal prediction beyond exchangeability. *Annals of Statistics*, *51*. doi:10.1214/23-AOS2276

Barbounis, T. G., and Theocharis, J. B. (2006). Locally recurrent neural networks for long-term wind speed and power prediction. *Neurocomputing*, *69*. doi:10.1016/j.neucom.2005.02.003

Breiman, L. (2001). Random forests. *Machine Learning*, *45*. doi:10.1023/A:1010933404324

Broomhead, D. S., and King, G. P. (1986). Extracting qualitative dynamics from experimental data. *Physica D: Nonlinear Phenomena*, *20*. doi:10.1016/0167-2789(86)90031-X

Cadenas, E., and Rivera, W. (2010). Wind speed forecasting in three different regions of Mexico, using a hybrid ARIMA-ANN model. *Renewable Energy*, *35*. doi:10.1016/j.renene.2010.04.022

Cai, H., Jia, X., Feng, J., Yang, Q., Hsu, Y. M., Chen, Y., and Lee, J. (2019). A combined filtering strategy for short term and long term wind speed prediction with improved accuracy. *Renewable Energy*, *136*. doi:10.1016/j.renene.2018.09.080

Cleveland, R. B., Cleveland, W. S., McRae, J. E., and Terpenning, I. (1990). STL: A Seasonal-Trend Decomposition Procedure Based on Loess (with Discussion). *Journal of Official Statistics*, *6*.

Croonenbroeck, C., and Ambach, D. (2015). A selection of time series models for short- to medium-term wind power forecasting. *Journal of Wind Engineering and Industrial Aerodynamics*, *136*. doi:10.1016/j.jweia.2014.11.014

Devi, A. S., Maragatham, G., Boopathi, K., Lavanya, M. C., & Saranya, R. (2021). Long-term wind speed forecasting—A review. Lecture Notes in Networks and Systems, Vol. 130. doi:10.1007/978-981-15-5329-5_9

DNV (2012). Windiness. *Report.* https://www.dnv.com/article/windiness-2022-240811.

Dragomiretskiy, K., and Zosso, D. (2014). Variational mode decomposition. *IEEE Transactions on Signal Processing*, *62*. doi:10.1109/TSP.2013.2288675

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*. doi:10.1214/aos/1013203451

Geurts, M., Box, G. E. P., and Jenkins, G. M. (1977). Time series analysis: forecasting and control. *Journal of Marketing Research*, *14*. doi:10.2307/3150485

Gibbs, I., and Candès, E. J. (2021). Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems*, *3*.

Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., and Yang, G. Z. (2019). XAI-Explainable artificial intelligence. *Science Robotics*, *4*. doi:10.1126/scirobotics.aay7120

Gwec. (2022). Global offshore wind report 2022. *Global Wind Energy Council*.

Huang, C. Y., Liu, Y. W., Tzeng, W. C., and Wang, P. Y. (2011). Short term wind speed predictions by using the grey prediction model based forecast method. *2011 IEEE Green Technologies Conference, Green 2011*. doi:10.1109/GREEN.2011.5754856

Huang, N., Xing, E., Cai, G., Yu, Z., Qi, B., and Lin, L. (2018). Short-term wind speed forecasting based on low redundancy feature selection. *Energies*, *11*. doi:10.3390/en11071638

Iea. (2022). World energy outlook 2022, IEA, Paris https://www.iea.org/reports/world-energy-outlook-2022, License: CC BY 4.0 (report); CC BY NC SA 4.0 (Annex A). *CC BY 4. 0 (Report); CC BY NC SA 4. 0 (Annex A)*.

Irnea. (2022). *Renewable power generation costs in 2021*.

Jensen, V., Bianchi, F. M., and Anfinsen, S. N. (2022). Ensemble conformalized quantile regression for probabilistic time series forecasting. *IEEE Transactions on Neural Networks and Learning Systems*. doi:10.1109/TNNLS.2022.3217694

Jiang, P., Wang, Y., and Wang, J. (2017). Short-term wind speed forecasting using a hybrid model. *Energy*, *119*. doi:10.1016/j.energy.2016.10.040

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., … Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, *2017-December*.

Kolle, O. (2008). Documentation of the weather station on top of the roof of the institute building of the max-planck-institute for biogeochemistry. Max-Planck-Institute for Biogeochemistry. https://www.bgc-jena.mpg.de/wetter/Weatherstation.pdf.

Liu, H., and Chen, C. (2019). Data processing strategies in wind energy forecasting models and applications: A comprehensive review. *Applied Energy*, Vol. 249. doi:10.1016/j.apenergy.2019.04.188

Lundberg, S. M., and Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, *2017-December*.

Malik, H., and Savita. (2016). Application of artificial neural network for long term wind speed prediction. *Conference on Advances in Signal Processing, CASP 2016*. doi:10.1109/CASP.2016.7746168

Mogos, A. S., Salauddin, M., Liang, X., and Chung, C. Y. (2022). An effective very short-term wind speed prediction approach using multiple regression models. *IEEE Canadian Journal of Electrical and Computer Engineering*, *45*. doi:10.1109/ICJECE.2022.3152524

Papadopoulos, H., Proedrou, K., Vovk, V., and Gammerman, A. (2002). Inductive confidence machines for regression. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *2430*. doi:10.1007/3-540-36755-1_29

Riahy, G. H., and Abedi, M. (2008). Short term wind speed forecasting for wind turbine applications using linear prediction method. *Renewable Energy*, *33*. doi:10.1016/j.renene.2007.01.014

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). 'why should i trust you?' explaining the predictions of any classifier. *NAACL-HLT 2016 - 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Demonstrations Session*. doi:10.18653/v1/n16-3020

Romano, Y., Patterson, E., and Candès, E. J. (2019). Conformalized quantile regression. *Advances in Neural Information Processing Systems*, *32*.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2020). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision*, *128*. doi:10.1007/s11263-019-01228-7

Shrikumar, A., Greenside, P., and Kundaje, A. (2017). Learning important features through propagating activation differences. *34th International Conference on Machine Learning, ICML 2017*, *7*.

Shukur, O. B., and Lee, M. H. (2015). Daily wind speed forecasting through hybrid KF-ANN model based on ARIMA. *Renewable Energy*, *76*. doi:10.1016/j.renene.2014.11.084

Steinberger, L. and Leeb, H. (2018). Leave-one-out prediction intervals in linear regression models with many variables. https://arxiv.org/abs/1602.05801.

Thakur, M., and Kumar, D. (2018). A hybrid financial trading support system using multi-category classifiers and random forest. *Applied Soft Computing Journal*, *67*. doi:10.1016/j.asoc.2018.03.006