Open University of Catalonia (UOC) Master's Degree in Data Science

MASTER'S THESIS

Area: 5

Enhancing Weather Analysis with Data Interpolation and Time Series Forecasting

---

Author: Alejandro González Barberá

Tutor: Sergio Iserte Agut

Professor: Albert Solé Ribalta

June 29, 2023

Credits/Copyright

FINAL PROJECT RECORD

Title of the project:	Weather Data Interpolation and Time Series Forecasting
Author's name:	Alejandro González Barberá
Collaborating teacher's name:	Sergio Iserte Agut
PRA's name:	Albert Solé Ribalta
Delivery date (mm/yyyy):	06/2023
Degree or program:	Data Science
Final Project area:	Area 5
Language of the project:	English
KeyWords:	Weather Analysis, Machine Learning, Data Interpolation

Quote

"Data is the new oil. It's valuable, but if unrefined, it cannot really be used." - Clive Humby

Acknowledgements

The successful completion of this project would not have been possible without the support and contributions from various individuals and organizations. The author would like to express their sincere gratitude to the following:

1. *Servei Informàtica* from Universitat Jaume I (UJI) for hosting and maintaining Shirka, the calculus node. Their provision of the necessary resources and infrastructure was instrumental in conducting this research.

2. Avamet and Inforatge for generously providing the weather datasets used in this study. Their valuable data has been instrumental in our analysis and model development.

The author extends their appreciation to all those who have directly or indirectly contributed to the successful completion of this project. Their support and collaboration have been instrumental in advancing our understanding of weather forecasting and water management.

Also, the author would like to extend his deepest appreciation to the tutor, Sergio Iserte, for the continuous support, valuable insights, and patience. His dedication and mentorship have been instrumental in shaping the direction and outcomes of this project.

Abstract

Weather analysis plays a crucial role in various domains, from agriculture to urban planning. However, accurate and localized predictions can be challenging in urban environments with limited weather station coverage. In this work, we propose a comprehensive workflow that combines dot rain coverage, interpolation, and machine learning models to address this issue. By establishing a network of weather stations strategically distributed across the city and utilizing their weather variables as input for the interpolation techniques, we generate interpolated data for the entire city grid. This approach enables us to fill the gaps in weather station coverage and provide accurate predictions for locations without direct measurements.

Subsequently, machine learning models are trained on the interpolated data to forecast various weather variables. We conducted extensive experiments and hyperparameter optimization to achieve accurate predictions with low evaluation loss. Furthermore, our models demonstrate transferability across different weather stations within the city, enabling localized predictions in previously unmonitored areas. The results highlight the effectiveness of our project in improving weather analysis capabilities in urban settings.

This work opens avenues for further research in applying these techniques to different regions with diverse weather conditions, ultimately enhancing decision-making processes in various sectors reliant on accurate weather predictions.

Keywords: Weather Analysis, Time Series, Machine Learning, Data Interpolation

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this section, the Master's Thesis overview is introduced which encompasses the general problem description, objectives, methodology, planning, hardware configuration, and thesis contributions.

1.1 General Problem Description

Currently, large amounts of data are being generated, especially in the industrial sector. However, most of this data is not utilized to improve decision-making for the stakeholders. In this project, we propose leveraging various data sources, such as meteorological weather stations

In this project, we address two main branches: spatial interpolation and weather variables study. The project aims to leverage meteorological data from various sources to improve decision-making . The specific objectives are as follows:

- Spatial Interpolation: This analysis includes examining spatial and temporal patterns, investigating the relationships between meteorological stations, and employing interpolation techniques to estimate rainfall values at unobserved locations. Accurate rainfall distribution information is crucial for water resource management, urban planning, and agricultural practices.

- Weather variables analysis: The project also focuses on predicting future weather conditions based on historical data and machine learning techniques. Accurate weather analysis enable stakeholders to plan activities, manage resources efficiently, and respond effectively to weather-related events.

1.1.1 Justification

Industrial Data Underutilization. The project addresses the prevalent issue of underutilization of data in the industrial sector. Despite the exponential growth in data generation, many stakeholders fail to fully leverage its potential for informed decision-making. This project aims to demonstrate the value of Data Science methods in unlocking hidden insights and improving decision-making processes. By harnessing advanced analytical techniques, such as spatial interpolation and machine learning, the project aims to optimize resource allocation and enhance overall efficiency.

Advancement of Weather Analysis. The project recognizes the need for advanced analytical tools in weather variables analysis. With the increasing availability of high-resolution weather data, traditional methods often struggle to handle the complexity and volume of information. By leveraging spatial interpolation and machine learning algorithms, this project aims to overcome these limitations and provide accurate and timely weather predictions. This advancement in weather analysis techniques can benefit a wide range of industries and sectors that heavily rely on weather information, including agriculture, transportation, energy, and disaster management.

Bridging the Gap between Data and Actionable Information. The project's multidisciplinary approach, combining spatial interpolation, machine learning, and data analysis, aims to bridge the gap between raw weather data and actionable information. By developing algorithms and models, the project seeks to transform vast amounts of data into valuable insights and predictions.

1.1.2 Motivation

This project is motivated by the recognition of the untapped potential of data in the industrial sector. Despite the availability of vast amounts of data, many organizations fail to fully leverage its value for informed decision-making. In the context of weather analysis, this project seeks to demonstrate the effectiveness of data science methods in improving decision-making processes and enhancing efficiency.

Another key motivation for this project is the increasing demand for accurate weather information across different industries. Sectors such as agriculture, transportation, logistics, and energy rely heavily on precise weather predictions to optimize their operations and mitigate risks. By developing advanced weather analysis techniques, the project aims to meet this growing demand and provide stakeholders with reliable and actionable insights. Ultimately, the

motivation is to enhance the resilience, efficiency, and sustainability of industries by leveraging the power of data science in weather variable analysis and prediction.

1.2 Objectives

The project aims to achieve the following objectives:

- Spatial interpolation: Create a grid of points within the study area by interpolating the measured weather variable values from meteorological stations. This interpolation process will provide a more precise and comprehensive representation of data distribution across the domain.
- Weather variables prediction: Develop a predictive model that can accurately forecast weather variables at any location within the study area. Machine learning techniques, including neural networks, will be employed to capture the complex relationships in the data and make reliable predictions.
- Correlation analysis: Investigate the correlations between weather variables. Various analytical approaches, such as linear correlation analysis and linear regression, will be employed to explore the nature and strength of the relationship between these variables. Advanced data analysis techniques will be also utilized to uncover complex patterns and relationships.
- Machine learning model comparison: Compare different machine learning models for weather forecasting. Cross-validation techniques and model evaluation metrics will be employed to assess the accuracy and efficiency of the models in terms of training time and prediction performance. The characteristics of the models, including complexity and bias-variance trade-off, will also be analyzed to gain insights into their behavior and generalization capability.

1.3 Methodology

The project is divided into two main blocks, as mentioned earlier. In the first block, the problem of interpolation at any point in the study area based on available meteorological station data is addressed. The methodology for this block involves the following steps:

1. Data preprocessing: The measured data from the meteorological stations will be cleaned and transformed to prepare them for use in the predictive model.

2. Grid point creation: A grid of points will be generated within the domain of the study area by interpolating the measured data from the meteorological stations.

3. Development of the predictive model: machine learning techniques will be used to construct a model that can predict at any location within the domain based on the meteorological station data.

4. Model evaluation: Various evaluation metrics, such as mean squared error , will be used to determine the accuracy and effectiveness of the developed model.

In the second block, the analysis of correlations between weather variables is addressed, and different machine learning models for the prediction of these are compared. The methodology for this block is as follows:

1. Correlation analysis: Different correlation analysis techniques will be used to analyze the relationship between the differente weather variables.

2. Development of predictive models: Different machine learning models will be compared for the forecasting task.

3. Evaluation and selection of the best model: Various cross-validation techniques and evaluation metrics will be used to determine which model is the most accurate and efficient in terms of training time and prediction performance.

4. Validation of the model: The selected model will be validated using the weather stations that were not included in the interpolation process. This validation will provide an unbiased assessment of the model's performance on unseen data and its generalization capability.

Figure 1.1 illustrates the overall workflow of the project, including data collection and preprocessing, model creation and optimization, interpolation, inference, evaluation and analysis.

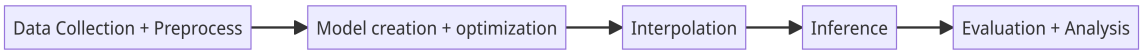


Figure 1.1: Workflow of the project.

1.4 Planning

The project is tracked with a Gantt chart, and the tool used to create it was Team Gantt. The Gantt chart can be seen in Figure 1.2. The project is divided into five main blocks corresponding to the PACs:

1. **Definition and planning of the project (1-12 March)**
    - Data collection from meteorological stations.
    - Definition of objectives, requirements, and methodology.
    - Project planning.
2. **Delve into the State-of-the-Art (13-26 March)**
    - Research the latest advancements in machine learning models for time series forecasting.
    - Explore different interpolation methods for creating a grid within the domain of the meteorological stations.
3. **Implementation (27 March - 28 May)**
    - Preprocess the data, including cleaning, normalization, and feature engineering, to prepare it for further analysis.
    - Create a grid within the domain of the meteorological stations using appropriate interpolation methods.
    - Analyze and compare different interpolation methods to determine their suitability for the task.
    - Explore and develop machine learning models specifically designed for time series forecasting.
    - Train and evaluate the performance of the machine learning models using the prepared grid data.
    - Validate the accuracy and reliability of the trained models through rigorous testing and comparison.
4. **Writing the manuscript pt 1 (29 May - 11 June)**
    - Writing the manuscript (Introduction, Methodology).
5. **Writing the manuscript pt 2 (12-25 June)**
    - Writing the manuscript (Experimental Results, Discussion, Conclusion).
    - Writing the manuscript (Abstract, Conclusions).
    - Preparing the presentation.
    - Reviewing and finalizing the manuscript.

6. **Defense preparation (26 June - 2 July)**
    - Rehearsing the presentation.
    - Finalizing the presentation slides.
7. **Defense (3-14 July)**

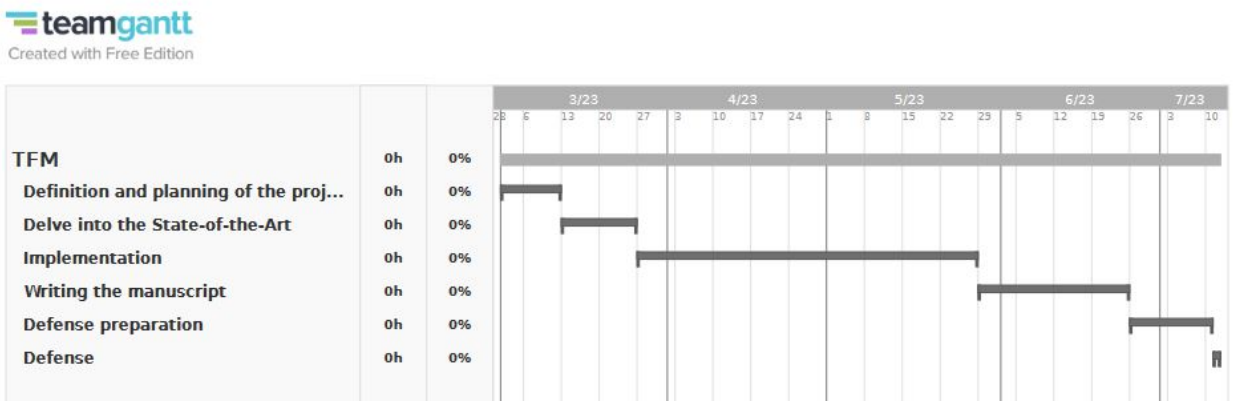


Figure 1.2: Gantt chart of the project.

1.5 Hardware Configuration

This section describes the hardware used for this project including a desktop computer and a high-performance computing cluster.

1.5.1 Desktop Computer

The baseline computer utilized for low-demanding tasks, such as data preprocessing, exploratory analysis, and plotting, was equipped with the following specifications:

- CPU: AMD Ryzen 7 5800X
- GPU: NVIDIA GeForce RTX 3060 Ti
- Memory: 32GB RAM

This configuration provided sufficient computational power to handle the preliminary tasks and generate visualizations.

1.5.2 High-Performance Computing Cluster

For the high-demanding tasks, such as model training and performance evaluation, a high-performance computing cluster was utilized. The cluster, provided by *Universitat Jaume I (UJI)*, offered advanced computing resources to accelerate the training process. The cluster configuration consisted of:

- CPUs: 2 x Xeon(R) Gold 5220R CPU @ 2.20GHz (24 cores each)
- Memory: 12 x 32GB DDR4 at 2933 MT/s (2666 MT/s)
- GPUs: 4 x Tesla V100-SXM2 32GB

The utilization of the cluster allowed for efficient training and evaluation of the time series prediction models, leveraging the computational power of GPU and high-performance servers.

1.6 Contributions

The main contributions of this work can be summarized as follows:

1. Introducing a comprehensive workflow for weather analysis that combines interpolation methods with time series prediction models.
2. Evaluating and comparing different interpolation techniques for estimating weather variables across a geographical region.
3. Developing and training a neural network model for time series prediction, considering various architectures and hyperparameters.
4. Validating the performance of the proposed framework using real-world weather data and demonstrating its effectiveness in generating accurate and spatially distributed weather forecasts.

By integrating these components, we aim to provide a holistic approach to weather forecasting that enhances the accuracy and applicability of predictions in diverse geographical regions.

Chapter 2

Background

In this section, we provide an introduction to the previous work conducted as part of the author's own Bachelor's thesis . This thesis serves as a foundational piece of research that has contributed to our current investigation. Additionally, we discuss the related work in the areas of Weather Data Analysis and Spatial Interpolation Techniques.

2.1 Previous Work

This work arises from the Final Degree Project named "Estudio de parámetros influyentes en el flujo de entrada de estaciones de depuración de aguas residuales" presented in UJI on the 22nd of June 2022.

2.1.1 Context

Water supply is an essential resource for human life, making efficient and sustainable management crucial. Wastewater treatment plants (WWTPs) are essential infrastructures that treat wastewater to remove contaminants before returning it to the environment. The proper functioning of these plants is fundamental to ensure water quality, compliance with environmental standards, and public health.

However, the management of WWTPs is affected by external factors such as climate and hydrological variability, which can generate unpredictable and unstable inflow rates. In particular, heavy rainfall can cause a sudden and significant increase in inflow rates, which can negatively impact plant operation and reduce wastewater treatment efficiency. Therefore, it is crucial to understand and predict how rainfall affects the inflow rates of WWTPs in order to optimize their management.

2.1.2 Presentation of Bachelor's Thesis

This project is based on the analysis of inflow rates in wastewater treatment plants (WWTPs), in conjunction with rainfall data collected from meteorological stations located within the same WWTPs. The objective is to predict the behavior of water inflow rates in the future using statistical and machine learning methods. Data from multiple treatment plants are available, which need to be studied both independently and collectively.

The main objective of this project is to provide useful information to the technicians of the WWTPs to improve the inflow rate prediction system throughout different seasons of the year. To achieve this, the following specific objectives must be pursued:

- Read and analyze inflow rate and rainfall data from at least one WWTP.
- Select the best predictive model and train it using the acquired data.
- Make future predictions about inflow rates during different seasons.
- Present statistical reports displaying various variable trends.

This project is divided into two main blocks: rainfall analysis and correlation with inflow rate data, and comparison of different predictive models for inflow rates.

In the first block, rainfall data collected from meteorological stations will be analyzed, and correlations with inflow rate data from the WWTPs will be established.

Regarding the second block, different predictive models will be compared using machine learning techniques to determine the model that best fits the data and is capable of making accurate predictions.

The expected outcome of this project is to obtain a predictive model that accurately predicts inflow rates in different WWTPs during various seasons. Additionally, it aims to provide useful information to the technicians of the WWTPs to enhance the inflow rate prediction system throughout different times of the year.

Furthermore, statistical reports showcasing various variable trends will be provided, and a user interface will be developed to facilitate easy and fast access to the results.

2.1.3 Results

Firstly, the correlation between inflow rates and rainfall collected by the meteorological stations was analyzed. After several analyses, it was deduced that there was no statistically significant correlation between the two variables. Additionally, a problem was identified in the dataset: a large imbalance between observations with and without rainfall.

Regarding the predictive models, three approaches were tested: a recurrent neural network with Long Short-Term Memory (LSTM) layers, Facebook's Prophet tool, and the XGBoost gradient boosting tree. All models were configured with the best hyperparameters found through the Optuna framework.

In the case of a non-rainy interval, the three models had similar accuracy, as shown in Figure 2.1. However, when a rainy period occurred, only XGBoost was able to generalize the trends, as depicted in Figure 2.2. This was due to the aforementioned data imbalance, as XGBoost and the ensemble model family are designed to handle data imbalances.

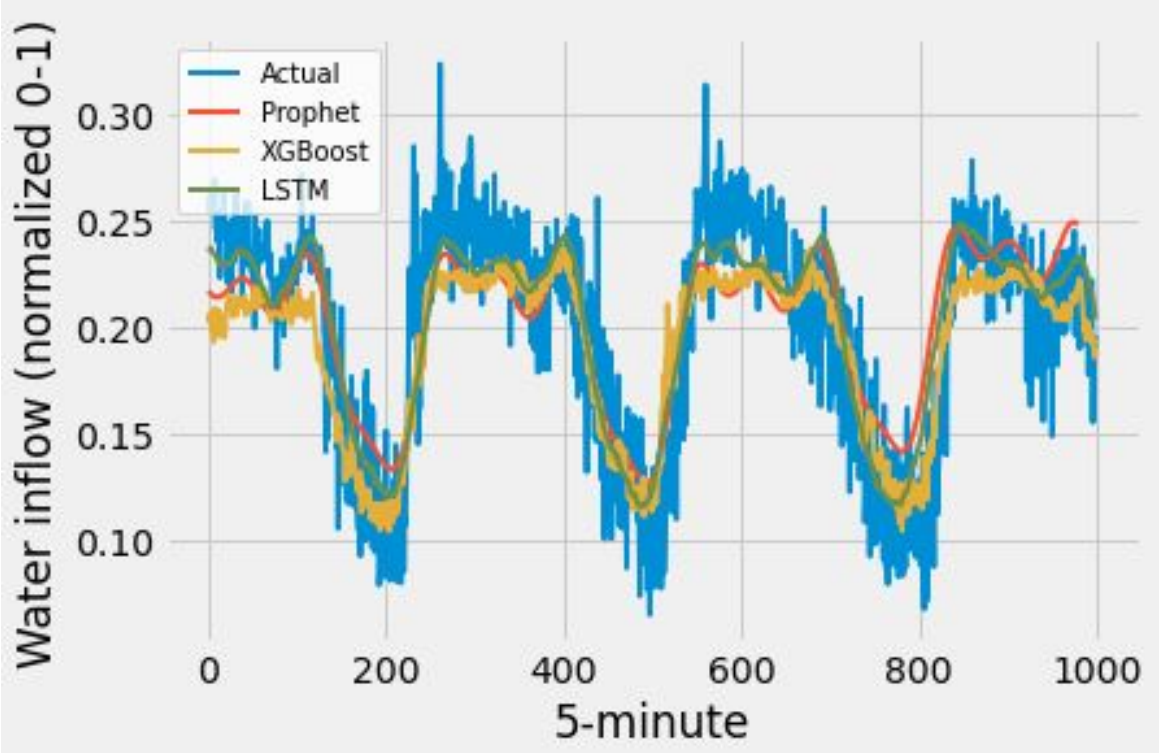


Figure 2.1: Comparison of predictions for non-rainy interval by the models.

2.1.4 Discussion

In conclusion, an analysis of the parameters influencing the inflow rates of WWTPs has been conducted. It has been demonstrated that there is no statistically significant correlation between rainfall and inflow rates, although it is evident that rainfall has an impact on increasing the inflow rates. Additionally, it has been observed that the inflow rates exhibit a certain periodicity over time, but this periodicity has been reduced in recent times due to a significant amount of rainfall occurring during atypical periods.

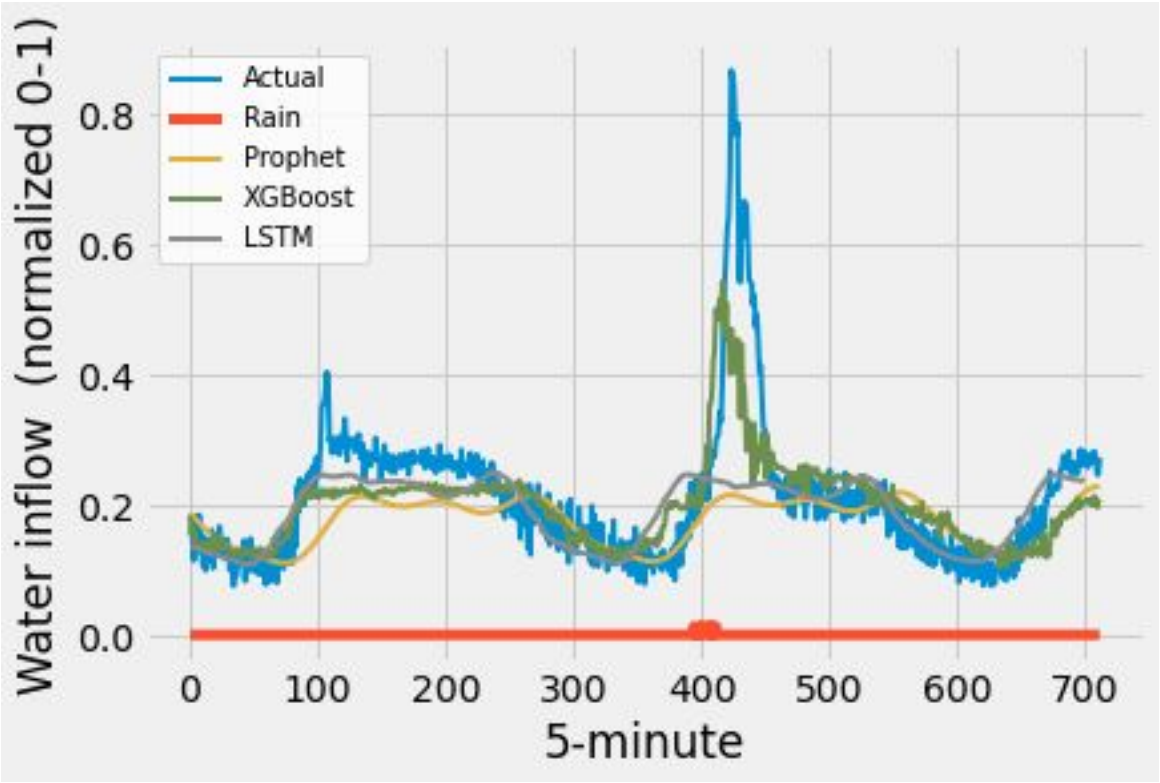


Figure 2.2: Comparison of predictions for rainy interval by the models.

Regarding the predictive models, it has been observed that for a non-rainy interval, the three models have similar accuracy, while for a rainy interval, only XGBoost is capable of generalizing the trends due to the significant data imbalance.

Therefore, it can be concluded that XGBoost is the model that best predicts the inflow rates in the analyzed dataset. However, further detailed studies are required to determine if the results hold in other situations and if better results can be achieved with other models or techniques.

2.2 Related Work

Weather analysis and spatial interpolation techniques play crucial roles in data-driven weather prediction. This section presents a review of relevant literature in the field, covering benchmark datasets, weather forecasting models, and spatial interpolation techniques.

2.2.1 Weather Data Analysis

In the study by Fathi et al. (1), a systematic review of big data analytics in weather forecasting is presented. The authors analyze various techniques, including machine learning and

deep learning approaches, used to process and analyze large-scale weather data for accurate predictions.

Transductive Long Short Term Memory (LSTM) (2) is proposed as an effective model for time-series weather prediction. The transductive LSTM leverages the temporal dependencies in weather data and achieves improved forecasting accuracy compared to traditional methods.

A deep learning-based fine-grained weather forecasting model is introduced by Trovati et al. (3). The model incorporates spatial and temporal features extracted from weather data to provide detailed predictions at a local level.

Temporal Convolutional Neural (TCN) networks are utilized for weather forecasting using time-series data from local weather stations (4). The TCN model captures long-term dependencies in weather data and achieves competitive forecasting accuracy.

The WeatherBench dataset (5) has emerged as a benchmark dataset for data-driven weather forecasting. It provides a comprehensive collection of global weather forecasts at various spatial and temporal resolutions, enabling researchers to develop and evaluate advanced forecasting models.

2.2.2 Spatial Interpolation Techniques

Spatial interpolation techniques are widely used to estimate weather variables at unobserved locations. In the study by Pellicone et al. (6), several interpolation techniques are applied to monthly rainfall data in the Calabria region, Southern Italy. The authors compare the performance of different methods, including kriging, inverse distance weighting, and radial basis functions, in capturing the spatial distribution of rainfall.

In the domain of groundwater analysis, the study by Bronowicka-Mielniczuk et al. (7) compares different interpolation techniques for determining the spatial distribution of nitrogen compounds. The authors evaluate methods such as ordinary kriging, inverse distance weighting, and spline interpolation, highlighting their strengths and limitations in estimating pollutant concentrations.

Spatial interpolation techniques are also applied to participatory sensing data for various applications. The work by Iqbal et al. (8) explores the use of different interpolation methods for estimating air pollution levels based on participatory sensing data. The authors compare techniques such as kriging, inverse distance weighting, and natural neighbor interpolation, providing insights into their performance in capturing spatial patterns.

Furthermore, the estimation of population exposure to fine particulate matter is investigated by Li et al. (9). The study explores spatiotemporal interpolation methods for estimating population exposure using fine particulate matter data in the contiguous U.S. The authors

propose novel approaches to improve the accuracy of exposure estimation and develop a real-time web application for visualizing the results.

Rainfall spatial estimations are reviewed by Hu et al. (10), encompassing various techniques from spatial interpolation to multi-source data merging. The authors discuss the strengths and limitations of methods such as kriging, geostatistical methods, and data assimilation techniques for achieving accurate and reliable rainfall spatial estimations.

In summary, the literature presents a diverse range of approaches for weather forecasting and spatial interpolation. These techniques contribute to the advancement of data-driven weather prediction models and provide valuable insights into the estimation of weather variables at unobserved locations.

Chapter 3

Weather Variables Analysis

This chapter presents a comprehensive study of the weather variables, focusing on the application of interpolation methods, grid creation, weather variables analysis models, and their subsequent evaluation and validation.

3.1 Data Augmentation

Data augmentation is a crucial step in our project, aimed at enhancing the dataset for our comprehensive analysis of the weather variables grid. By employing various interpolation methods, we augment the available data to generate additional samples. This process allows us to expand the coverage and granularity of our dataset.

Interpolation serves as a powerful data augmentation technique, enabling us to fill in the gaps and estimate values at unobserved locations. Through this approach, we can create a denser grid representation of the weather variables, providing a more detailed understanding of their behavior across the study area.

The augmented dataset, enriched by interpolation, empowers us to train and evaluate our models on a more comprehensive and representative sample. This allows us to gain insights into the predictive performance of the models, assess the robustness of the interpolation techniques, and analyze the spatiotemporal relationships between the weather variables.

Initial dataset sceneario. The dataset used in this study comprises observations collected from multiple weather stations. These stations play a crucial role in providing us with valuable information about the weather conditions in different parts of the city of Castellón de la Plana, Spain. The following weather stations were included:

1. Tetuan, with coordinates 40.0073, -0.04727.

2. Borriol Gaeta, with coordinates 40.0169, −0.125.

3. Centre Urbà, with coordinates 39.9851, −0.06071.

4. Palau Festa, with coordinates 39.9746, −0.0314.

5. Port, with coordinates 39.9572, 0.0067.

6. Private source near WWTP.

7. Sos Baynat, with coordinates 39.98595, −0.0297.

8. Capitol, with coordinates 39.9830, −0.0367

9. Ribalta, with coordinates 39.9885, −0.0393.

These weather stations are strategically distributed across the city to capture the spatial variability of weather conditions. Each station records various meteorological variables, such as rainfall, temperature, humidity, and wind speed, providing us with a comprehensive dataset for analysis. Note that the private source is a weather station from a private person which do not want to publicly specify the coordinates, but it is assumed to be really close to the WWTP.

Weather station	**Begin**	**End**	**N obs**	**Interval**	**Lack of data**
Private_source	2013-04-23	2023-02-28	885,687	5-min	2017, 2019, 2020
Borriol_gaeta	2019-01-01	2021-12-31	157,617	10-min	-
Capitol	2021-04-08	2021-12-31	67,916	5-min	-
Centre_urba	2017-06-28	2023-01-13	174,951	30-min	-
Palau_festa	2017-12-29	2022-03-30	143,957	30-min / 10-min	-
Port_cs	2016-11-17	2023-01-13	171,717	30-min / 10-min	-
Ribalta	2021-11-13	2021-12-22	4801	30-min / 10-min	-
Sos_baynat	2019-01-01	2021-12-31	156,886	10-min	-
Tetuan	2017-11-19	2023-01-13	180,367	30-min / 10-min	-

Table 3.1: Description of the initial dataset

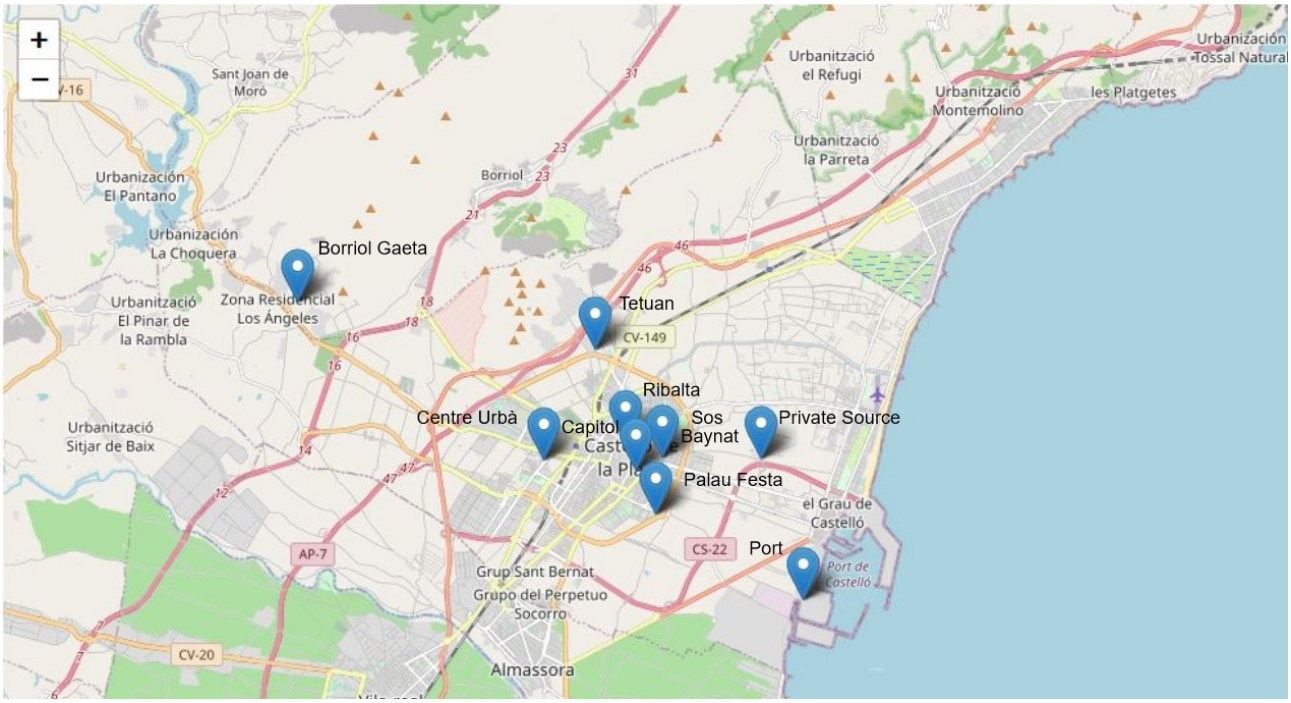


Figure 3.1: Location of meteorological stations in the city of Castellón

However, during the initial examination of the data (Table 3.1), it was observed that the Ribalta and Capitol stations had a limited number of data points compared to the other stations. This could potentially introduce inaccuracies and bias in our analysis if these stations were included in the initial dataset.

To ensure the reliability and accuracy of our findings, we made the decision to exclude the Ribalta and Capitol stations from the initial dataset. While this reduces the overall coverage of our data, it is a necessary step to maintain the integrity of our analysis. By focusing on the stations with a more extensive collection of observations, we can derive more robust insights and make reliable predictions based on the available data.

It is important to note that the removal of these stations will not have a significant impact on the subsequent analysis and processing of the data. Figure 3.1 shows the location of the meteorological stations.

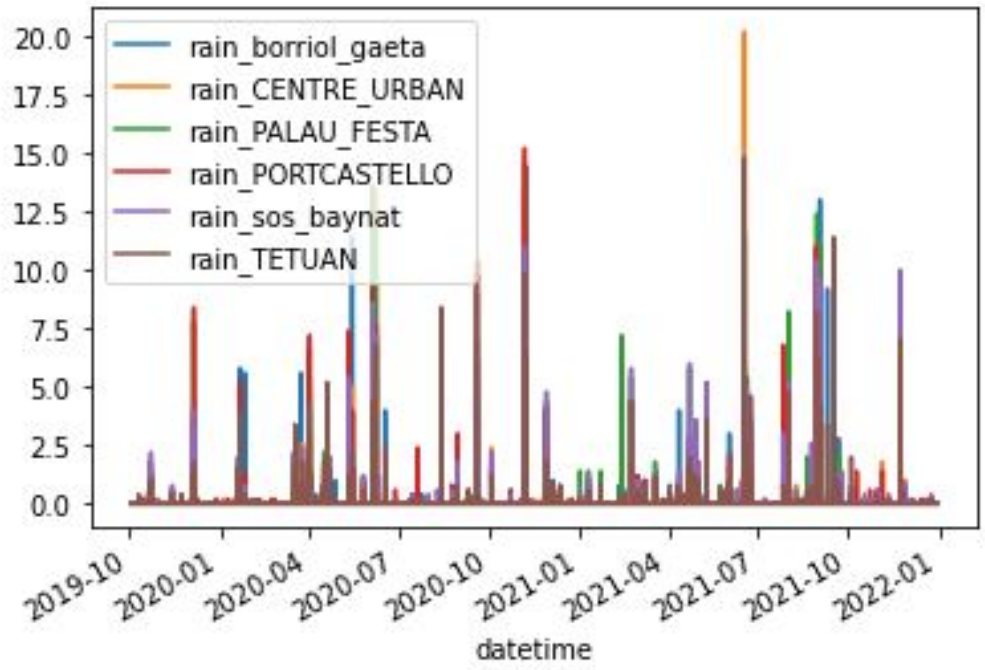


Figure 3.2: Representation of rainfall in the different meteorological stations.

Firstly, a study of the rainfall patterns in the area is going to be conducted, we have processed the collected data to create a comprehensive rainfall dataset. The resulting dataset is depicted in Figure 3.2. This dataset provides valuable insights into the distribution of rainfall across the selected weather stations.

In addition, we have examined the correlations between the different rainfall sources by constructing a correlation matrix, as shown in Figure 3.3. This matrix enables us to explore the relationships among the stations and identify any stations that may have a stronger influence on the overall rainfall dataset.

Upon inspection of the correlation matrix, we observe the highest correlation between the Tetuan and Centre Urbà stations, indicating a potential similarity in the climatic conditions between these two areas. Furthermore, the Palau Festa station exhibits strong correlations with the Centre Urbà and Sos Baynat stations, suggesting a potential relationship in rainfall patterns among these stations.

The analysis of the rainfall dataset and correlation matrix provides valuable insights into the interplay between different weather stations and their impact on the overall rainfall patterns in the study area. These findings will inform our subsequent analysis and modeling efforts in forecasting rainfall.

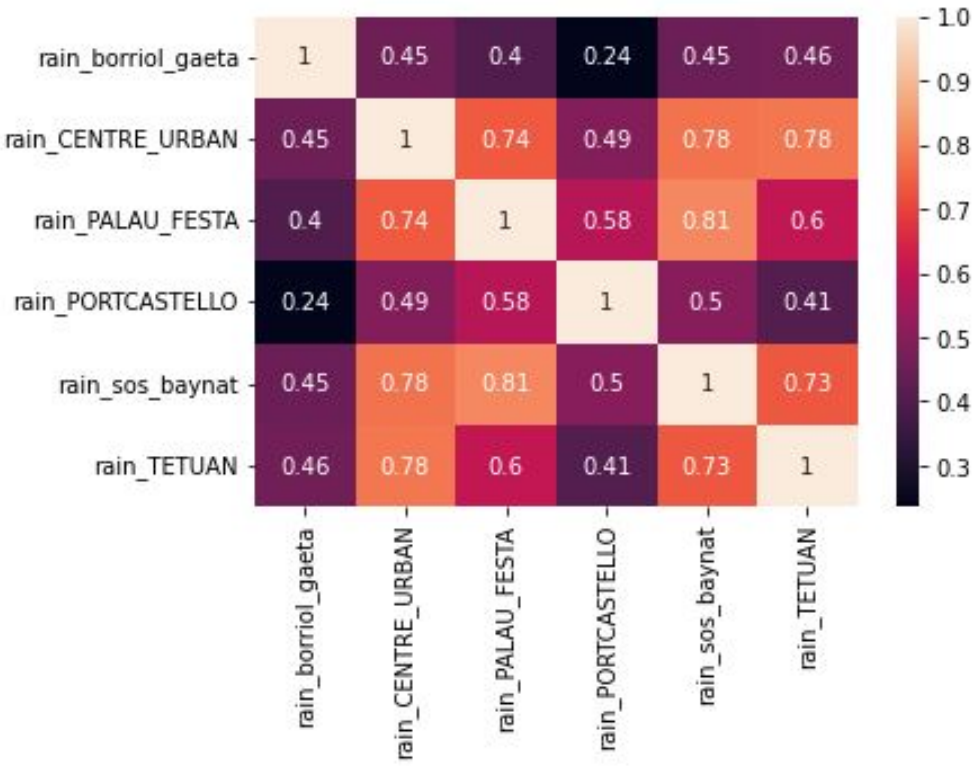


Figure 3.3: Correlation between rainfall collected from different meteorological stations.

3.1.1 Interpolation Techniques Comparison

As mentioned earlier, one of the objectives of this project is to calculate the different weather variables at several locations in the city of Castellón using interpolation techniques. In this initial use case the chosen variable is the rainfall but later on, the other variables are going to be calculated as well. Interpolation is a process by which the value of a function at a point is estimated based on the known values at nearby points. In this case, the dataset from the meteorological stations distributed throughout the city will be used to estimate the amount of rainfall at points where direct measurements are not available.

In order to validate the interpolations, data from the private source and two meteorological stations that were removed from the initial dataset due to their limited amount of data will be used. Different interpolation methods will be applied, and the results will be compared to determine which one is the most accurate.

In this project, interpolation methods from the `scipy.interpolate`[1] library have been used, which provides various options for interpolating data in multiple dimensions. Specifically, three methods have been employed: `Rbf`, `NearestNDInterpolator`, and `interp2d`:

[1] https://docs.scipy.org/doc/scipy/reference/interpolate.html

1. The `Rbf` method, short for Radial basis function, is a technique that uses radial basis functions as a basis for interpolation. These functions are used to fit a surface to the known data points. This method is useful for interpolating in multiple dimensions and can handle noisy data.

2. `NearestNDInterpolator` is a method that uses nearest-neighbor interpolation in multiple dimensions. This method finds the closest value to each point in the interpolation grid and uses it as the interpolated value. It is suitable for data that does not exhibit a clear trend and for high-dimensional spaces.

3. `interp2d` is a method specifically for 2D interpolation. It uses a bilinear interpolation technique to fit a surface to the known data. This method may be less accurate than other methods but is faster and less prone to oscillations.

The use of other interpolation methods, such as Lagrange or Newton, has been discarded as these methods are primarily suitable for interpolation in one dimension. Additionally, it has been empirically verified that Lagrange does not provide accurate results in this project.

As mentioned earlier, to evaluate the performance of the interpolations, rainfall data from three different locations has been used: private source, Capitol, and Ribalta because they were excluded from the interpolation. The metric used to measure the error in the interpolation is the widely known Mean Squared Error (MSE). The results are presented in Table 3.1.1.

Location	RBF	interp2d	nearest
WWTP	0.1889	0.8601	**0.1481**
Capitol	**0.1288**	0.1355	0.1427
Ribalta	0.1227	**0.0474**	0.1411

Table 3.2: MSE of different interpolation methods

Looking at the Mean Squared Error (MSE) values for each location, we can see that the RBF method outperforms the other two methods, interp2d and nearest, in terms of accuracy. For the WWTP location, the RBF method has an MSE of 0.1889, while the other methods have higher MSE values (0.8601 for interp2d and 0.1481 for nearest). Similarly, for the Capitol location, RBF has the lowest MSE of 0.1288 compared to the other methods (0.1355 for interp2d and 0.1427 for nearest).

Furthermore, when considering the Ribalta location, it is important to note that it has a relatively small dataset. Despite this limitation, the RBF method still performs reasonably well with an MSE of 0.1227, while the interp2d method has a higher MSE of 0.0474 and the nearest method has an MSE of 0.1411.

Based on these results, the RBF interpolation method demonstrates better overall performance in terms of accuracy compared to interp2d and nearest methods. It provides more reliable and precise estimates for locations with larger datasets, while also handling locations with smaller datasets relatively well. Therefore, considering its superior performance and robustness, the RBF method is the preferred choice for interpolating the rainfall data in this study.

Overall, the interpolation methods provide reasonable estimates of rainfall at locations where direct measurements are not available. However, it is important to note that interpolation introduces some level of uncertainty, and the choice of method should be based on the specific requirements and characteristics of the data.

3.1.2 Rain Dots Across the City

In order to conduct comprehensive analysis and modeling tasks, an evenly distributed grid of points was created across the city. The grid, consisting of 10 x 10 rows and columns, covers the entire study area and incorporates validation points corresponding to other weather stations. This design ensures a comprehensive coverage of the city, enabling accurate and representative analysis of the weather variables.

To visualize the distribution of the grid of points across the city refer to Figure 3.4, where the blue dots represent the created points and the red dots represent the validation weather stations.

By incorporating the points corresponding to other weather stations in the grid, we incorporate data from diverse sources and locations for validation. This inclusion enhances the robustness of the evaluation process and improves the reliability of the results obtained from the models.

The grid of points serves as a spatial framework for various analyses and modeling tasks. It facilitates the interpolation of rainfall data, enabling a detailed representation of rainfall distribution across the city. Furthermore, it allows for the comparison of different machine learning models and the evaluation of their performance in predicting weather variables at each point within the grid.

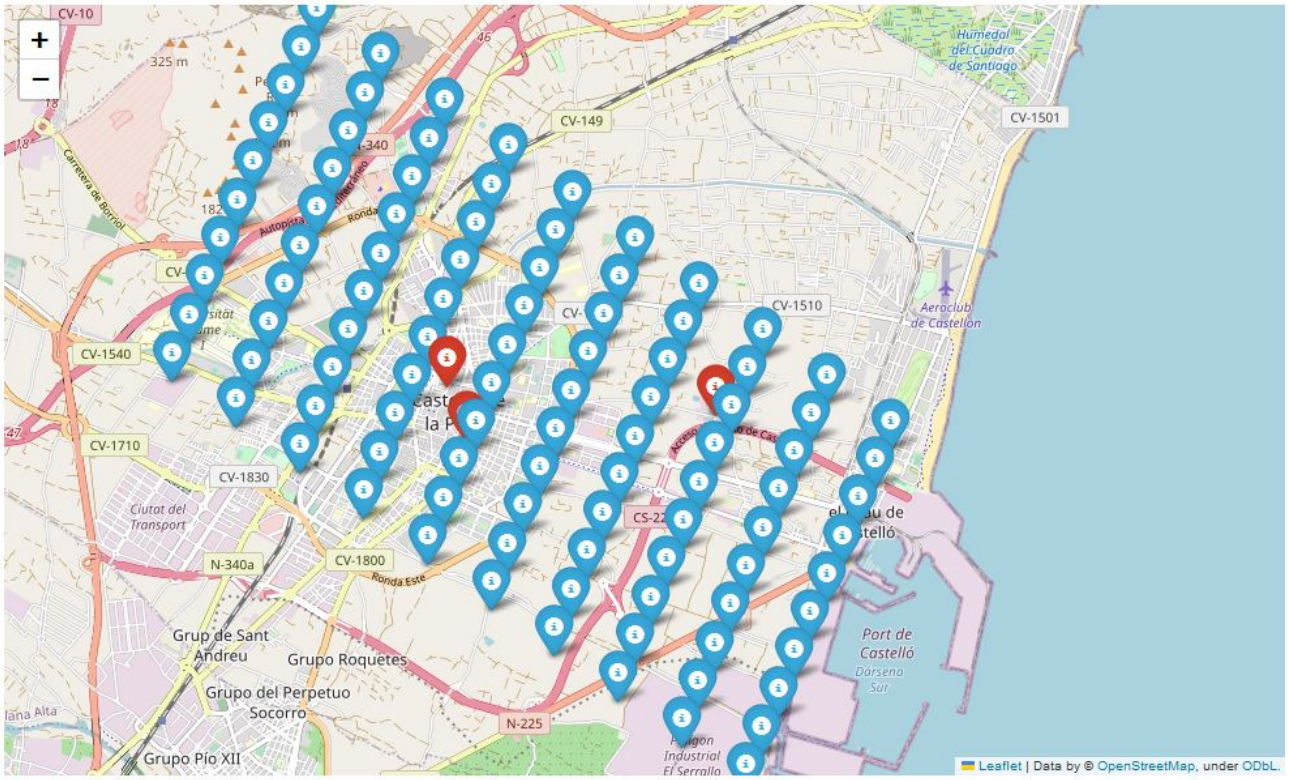


Figure 3.4: Grid of dots covering the city.

Overall, the grid of points provides a systematic and comprehensive approach to analyze and model the weather conditions in the city. It ensures the inclusion of validation points from other weather stations, enhancing the accuracy and reliability of the analysis.

3.1.2.1 Model: Weather Stations and Rainfall Interpolation

To build a predictive model for rainfall estimation, we leverage data from the six weather stations located across the city, as well as the interpolated rainfall values obtained for the rain dots. The combination of these data sources allows us to create a comprehensive dataset for training and evaluating the model.

Dataset Creation The dataset is constructed by gathering rainfall measurements from the six weather stations and the corresponding interpolated rainfall values for the rain dots. Each data sample in the dataset consists of the rainfall measurements from the weather stations as input and the interpolated rainfall value for a specific rain dot as the output.

The dataset is organized in a tabular format, with each row representing a data sample. The columns of the dataset include the rainfall measurements from each weather station, as

well as the target variable, which is the interpolated rainfall value for the corresponding rain dot.

Model Architecture For the development of the predictive model, we utilized a machine learning approach. The model architecture, as depicted in Figure 3.5, illustrates the structure and connections between the input layer, hidden layers and the output layer. In this particular case, the model consists of a sequential stack of layers, including dense layers. The simplicity of the problem allowed us to employ a straightforward architecture without extensive optimization.

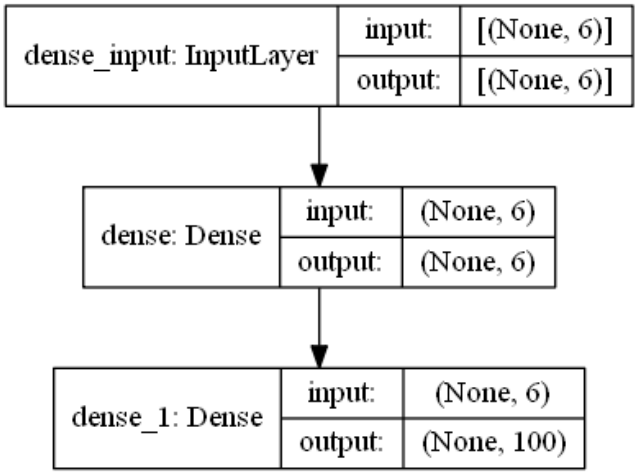


Figure 3.5: Model architecture for rainfall estimation.

Although the model we used is a basic one, it still provides a valuable framework for estimating rainfall. The choice of layers and activation functions can be adapted and expanded based on the complexity and requirements of the specific problem. In the case of our model, it consists of two dense layers: the input layer with 6 nodes and the output layer with 100 nodes. However, it's important to note that the model architecture can be further optimized and adjusted to improve performance if necessary.

Training and Evaluation The model is trained using the constructed dataset, utilizing appropriate algorithms and optimization techniques. The training process involves iteratively adjusting the model's parameters to minimize the difference between the predicted rainfall values and the actual interpolated rainfall values.

During the training process, the model's performance is continuously evaluated using mean squared error (MSE) to assess the accuracy of the predictions. The evaluation metric provide insights into how well the model captures the rainfall patterns and its ability to generalize to unseen data.

In Figure 3.6, a visualization of the model's error history and validation performance can be observed. This plot illustrates the changes in the error metric during the training process and the validation performance across different epochs. In addition, as the validation loss is higher than the loss during almost all the training, we can say that the model is not overfitting.

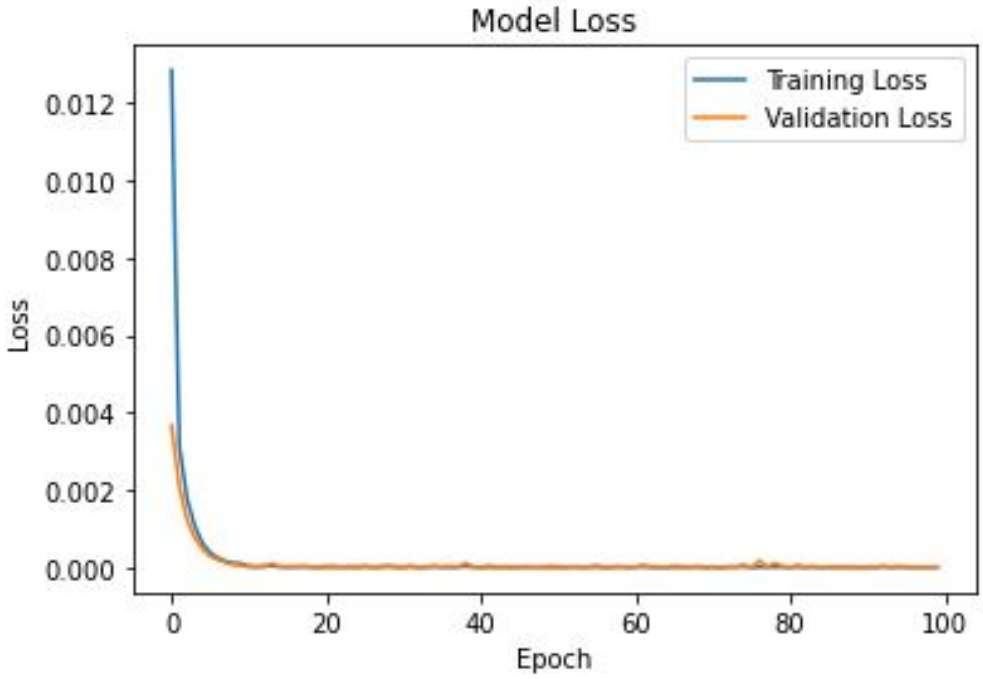


Figure 3.6: Model architecture for rainfall estimation.

The model's performance is not solely evaluated based on the training dataset, but also on a separate validation dataset. This validation dataset consists of rain dots for which rainfall measurements are available but were not used in the training process. Evaluating the model on the validation dataset helps assess its ability to generalize to new and unseen rain dots.

To illustrate the model's performance, Figure 3.7 presents a plot of the predicted rainfall values versus the actual rainfall values from the validation dataset. The plot reveals a nearly diagonal line, indicating a close correspondence between the predicted and actual values. This alignment demonstrates the model's ability to accurately estimate rainfall based on the provided inputs.

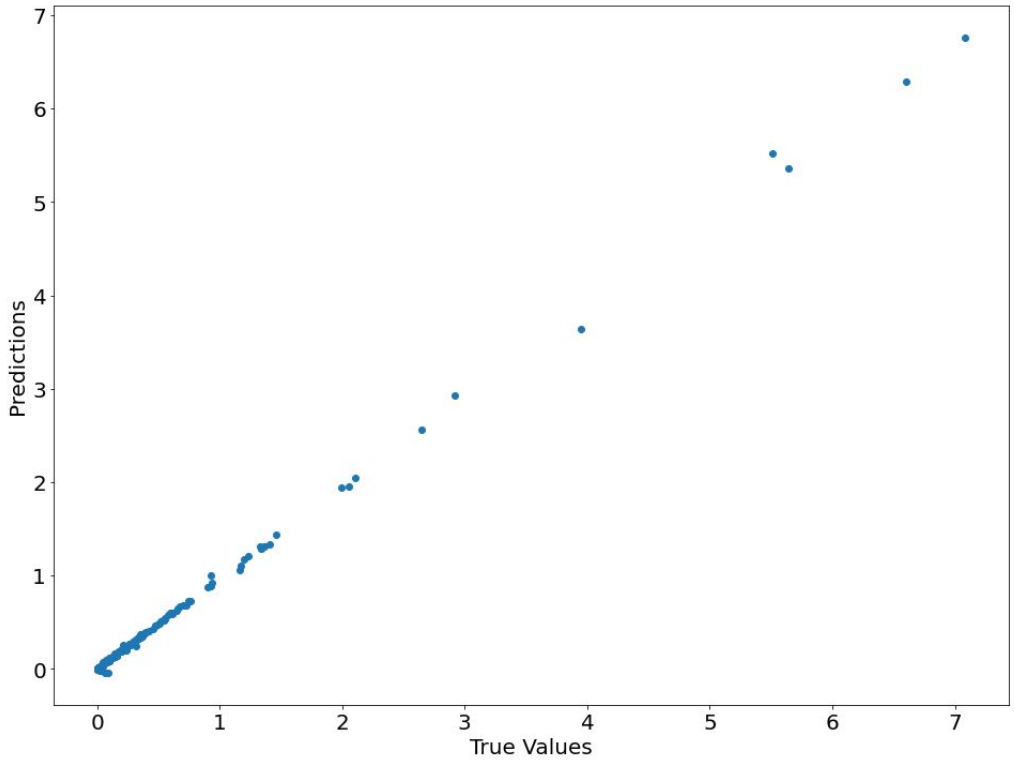


Figure 3.7: Comparison of predicted and actual rainfall values from the validation dataset.

The alignment of the data points along the diagonal line suggests that the model has achieved good results in capturing the underlying patterns and relationships within the rainfall data. This indicates that the model has successfully learned the necessary patterns from the training dataset and can effectively generalize to new rainfall data points.

3.1.2.2 Correlation Analysis: Rainfall in Grid of Dots and WWTP Inflow

In order to further analyze the relationship between rainfall in the grid of dots and the inflow water of the WWTP, we conducted a correlation analysis. This analysis aims to determine whether there is a significant correlation between the rainfall patterns observed in the grid of dots and the amount of water flowing into the WWTP.

The correlation analysis between the rainfall in the grid of dots and the WWTP inflow reveals a high correlation coefficient as 0.006167. This value indicates a very weak correlation between the rain dots and inflow water. A correlation coefficient close to 0 suggests that there is little to no linear relationship between the rainfall in the grid of dots and the WWTP inflow.

This lack of correlation suggests that other factors might influence the inflow water of the WWTP, such as underground water sources, drainage systems, or external factors not captured by the rainfall measurements in the grid of dots. Therefore, it is crucial to consider additional

data sources and factors when studying and predicting the water inflow to the WWTP or related hydrological processes.

By understanding the absence of correlation between rainfall in the grid of dots and the WWTP inflow, we gain valuable insights into the complex dynamics of water systems and the need for a comprehensive and multi-dimensional approach when studying and modeling such phenomena.

3.2 Time series Forecasting Model

3.2.1 Problem Statement

The problem we aim to address is the prediction of time series data for various weather variables, including humidity, temperature, pressure, wind velocity, and wind direction. Accurate predictions of these variables can have significant implications for a wide range of applications, such as agriculture, energy management, and urban planning.

Among the available weather data, we have selected these variables based on their relevance and potential impact on the aforementioned applications. Humidity, temperature, pressure, wind velocity, and wind direction play crucial roles in determining weather conditions and understanding the environment.

To approach this problem, we will collect historical data from a selected weather station that provides comprehensive and reliable measurements of the chosen weather variables. The selection of the weather station will be based on factors such as data quality, availability of the desired variables, and geographical relevance.

The collected data will then be preprocessed to handle missing values, outliers, and any other data quality issues. We will split the dataset into training, validation, and testing sets to evaluate and validate the performance of the prediction model.

Next, we will employ machine learning methods such as neural networks to build a predictive model. The model will be trained on the historical data to learn the patterns and dependencies among the weather variables. Once trained, the model will be capable of making accurate predictions of the weather variables for a specific time horizon, such as a one-day forecast.

By evaluating the model's performance on the validation and testing datasets, we can assess its accuracy and reliability in predicting the weather variables. The insights gained from the trained model can provide valuable information about the dynamics and relationships between the selected weather variables over time.

Also, to further enhance the accuracy and coverage of our weather predictions, we will incorporate an interpolation technique in conjunction with the trained model. By leveraging

the interpolated data generated from the selected weather stations distributed across the city, we can fill in the missing values and extend the predictions to the entire city grid. This interpolation step enables us to obtain more comprehensive and localized forecasts for each point within the grid. By assessing the performance of the model on the interpolated dataset and comparing it with the actual weather measurements, we can evaluate the effectiveness of the interpolation approach and ensure the reliability of the predictions across the entire city. This combined interpolation and prediction process strengthens the spatial representation of the weather variables and enhances the overall accuracy and usefulness of our forecasting model.

Overall, this approach will enable us to develop a model that can forecast the future values of humidity, temperature, pressure, wind velocity, and wind direction throughout an area based on historical data, allowing for improved decision-making and planning in various domains reliant on accurate weather information.

3.2.2 Data Preprocess

The TETUAN weather station is selected as the source of data for feeding the forecasting model. This station is chosen because it provides the most extensive dataset compared to other stations. The analysis focuses on predicting time series for variables such as humidity, temperature, pressure, wind velocity, and wind direction.

To tackle this problem, a dataset transformation is performed to create a supervised learning needed. The objective is to create a model that can predict the time series of these variables for a one-day period. To achieve this, a moving window approach is employed, where both input and output sequences are created. This creates a multi-step, multi-target, and multi-output problem.

The provided code snippet demonstrates the data transformation process. The function takes the original data, the number of input timesteps, and the number of output timesteps. It returns two arrays, X (data) and y (target), which represent the input and output sequences, respectively.

Before the transformation, the data is normalized using the *StandardScaler*[2] from the scikit-learn library. This ensures that the variables have similar scales and facilitates during all the process. The function then initializes the X and y arrays with appropriate shapes based on the number of samples, input steps, output steps, and the number of variables.

Next, a loop fills the X and y arrays by extracting the corresponding sequences from the original data. The sequences are flattened and stored in the arrays.

By utilizing this dataset transformation process, the time series data from the TETUAN

[2] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

weather station is prepared for training a predictive model. The transformed data will serve as the input for the model, enabling it to forecast the values of the selected variables for future time steps.

Please note that the code provided is an excerpt for illustrative purposes and should be adapted to the specific implementation.

```
def create_supervised_dataset(data, input_steps, output_steps):
    # Convert to numpy array
    data_array = data.values

    # scale data
    scaler_x = StandardScaler()
    data_array = scaler_x.fit_transform(data_array)

    # Compute number of variables and samples
    n_vars = data_array.shape[1]
    n_samples = data_array.shape[0]

    # Initialize X and y arrays
    X = np.zeros((n_samples - input_steps - output_steps + 1,
        input_steps * n_vars))
    y = np.zeros((n_samples - input_steps - output_steps + 1,
        output_steps * n_vars))

    # Fill X and y arrays
    for i in range(X.shape[0]):
        X[i, :] = data_array[i:i+input_steps, :].flatten()
        y[i, :] = data_array[i+input_steps:
                  i+input_steps+output_steps, :].flatten()

    return X, y
```

To illustrate the data transformation process, let's consider an initial DataFrame with the following structure:

Index	Temperatura	P_atmosferica	Humedad	Dir_viento	Vel_viento
0	12.8	1021.5	69.0	270.0	1.6
1	12.0	1021.1	69.0	225.0	1.6

```
2     13.9        1021.0        59.0    292.5      3.2
3     11.4        1021.1        67.0    337.5      1.6
4     13.1        1021.0        60.0    270.0      3.2
...   ...         ...           ...     ...        ...
n     9.9         1025.3        90.0    247.5      4.8
```

This DataFrame represents the weather variables of temperature, atmospheric pressure, humidity, wind direction, and wind velocity over time.

Following the workflow depicted, we perform the data transformation using a moving window approach. The goal is to create input and output sequences for supervised learning, where the model can predict the time series of these variables for a one-day period.

The transformation process involves normalization using the *StandardScaler* from the scikit-learn library, ensuring that the variables have similar scales.

The resulting transformed data will have the following shapes:

1. Shape of X (input sequences): (n, input timesteps), where n is the number of cases (number of rows in the original DataFrame).

2. Shape of y (output sequences): (n, output timesteps), representing the predicted values for the variable.

In the following subsections, we perform a comprehensive analysis of the data collected from Tetuan weather station. We focus on five key variables: temperature, humidity, pressure, wind velocity, and wind direction. The analysis includes examining the distributions of each variable, exploring their correlations, comparing variables against each other, and studying the tendencies over time.

3.2.3 Distributions of Variables

To gain an understanding of the data distribution for each variable, we examine their statistical properties. We analyze the following distributions depicted in Figure 3.8:

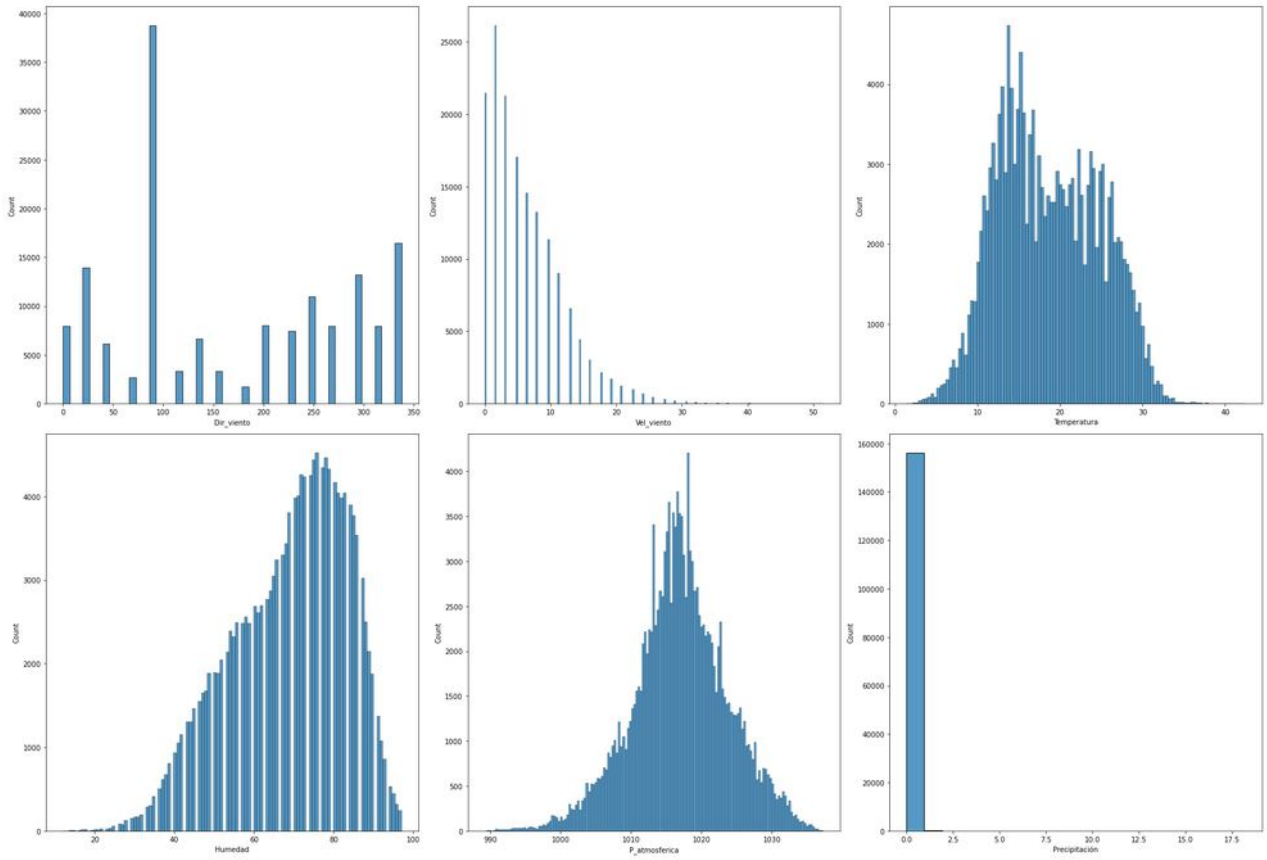


Figure 3.8: Distribution of each weather variable.

- Temperature (degrees Celsius): We investigate the distribution of temperature measurements to understand the range of temperatures recorded at the weather station. This analysis provides insights into the typical temperature patterns observed. The temperature distribution exhibits a bell-shaped curve with two maxima around 15 and 25 degrees Celsius, indicating that these temperatures are frequently observed.

- Humidity (Relative Humidity): We explore the distribution of humidity measurements to determine the humidity levels experienced at the weather station. Understanding the distribution helps in identifying the prevailing humidity conditions. The humidity distribution follows a beta distribution, allowing us to characterize the humidity levels and their likelihood of occurrence.

- Pressure (milibars): We analyze the distribution of pressure measurements to gain insights into the atmospheric pressure variations at the weather station. This information aids in understanding the pressure patterns and their potential impact on weather phenomena. The pressure distribution approximates a normal distribution centered around

1018 milibars, providing an overview of the typical atmospheric pressure at the station.

- Wind Velocity (kilometers per hour): We examine the distribution of wind velocity measurements to assess the strength and intensity of wind at the weather station. Analyzing the distribution helps in identifying different wind velocity categories and their frequency of occurrence. The wind velocity distribution follows a log-normal distribution, indicating that lower wind velocity values are more common, while higher wind velocity values occur less frequently.

- Wind Direction (degrees): We investigate the distribution of wind direction measurements to understand the prevailing wind patterns at the weather station. This analysis provides insights into the dominant wind directions and their frequencies. The wind direction distribution is uniformly distributed, but with a higher impact in the 100-degree range, indicating that wind frequently blows from that direction.

Please note that the distributions mentioned in the text (log-normal, beta, and normal) are theoretical representations based on the provided information. The actual distribution of the weather variables may vary depending on the specific data collected at the weather station. This distribution study help us to understand the balance of the data, which is well-balanced in all cases but in rainfall.

3.2.4 Correlations

To explore the relationships between the variables, we examine their correlations. Understanding the correlations between temperature, humidity, pressure, wind velocity, and wind direction can provide insights into the dependencies and interactions among these weather parameters.

To explore the relationships between the variables, we can start by analyzing the correlation matrix shown in Figure 3.9. This provides insights into the relationships between the variables. For example, we can observe that the highest correlation is between humidity and wind velocity, indicating a potential connection between these two factors. However, the correlations between the variables are generally low, suggesting that there is no strong linear relationship among them. After the relationships exploration, we should consider adding all of the variables but the rainfall as features on the model creation.

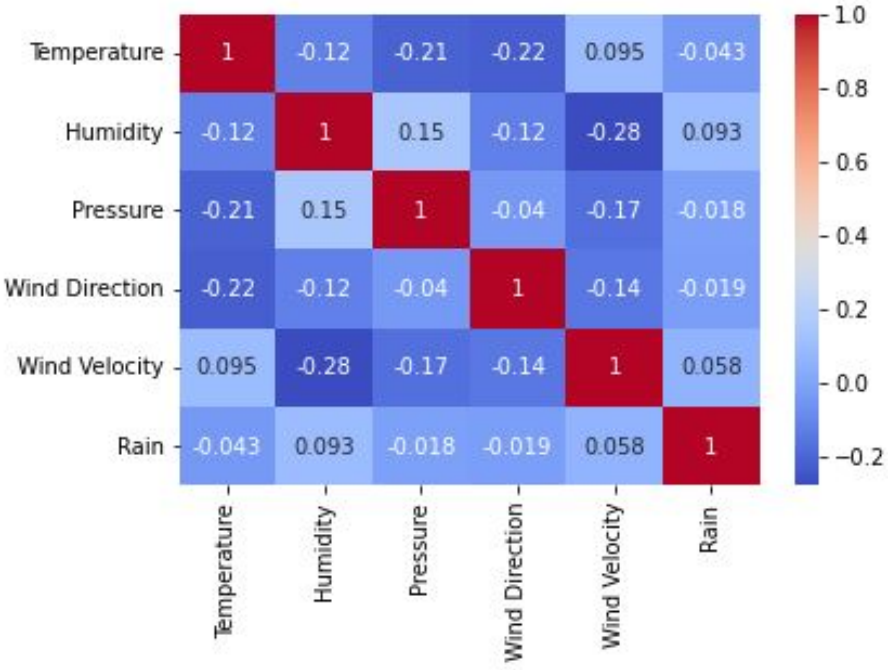


Figure 3.9: Correlation matrix between weather variables.

Next, we perform variable comparisons to explore the potential patterns and relationships between the weather parameters. Among these comparisons, the most interesting ones are between precipitation and other variables depicted in Figure 3.10. For instance, when it rains, we can observe that the temperature ranges from 10 to 20 degrees Celsius, humidity tends to be higher than 80%, wind velocity is less than 20 km/h, and atmospheric pressure is below 1020 milibars. These comparisons provide valuable insights into the conditions associated with rainfall occurrences.

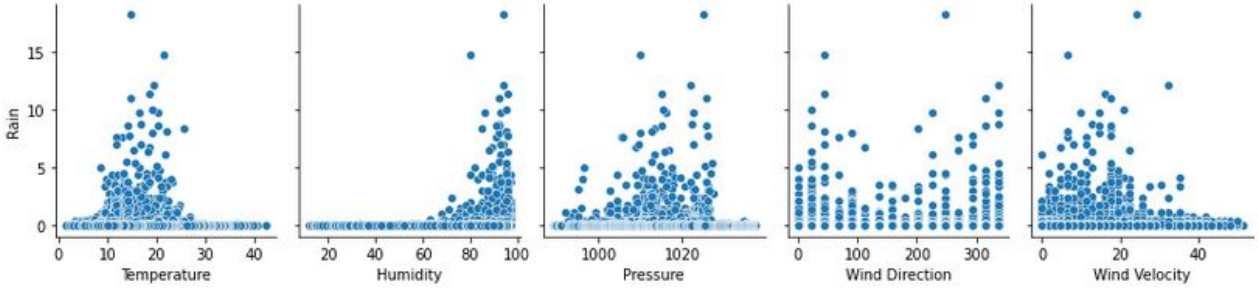


Figure 3.10: Correlation matrix between rain and weather variables.

By conducting these variable comparisons, we can uncover meaningful connections and dependencies between the weather parameters.

3.2.5 Tendencies Over Time

To gain insights into long-term patterns or trends, we examine the tendencies of the variables over time using the available historical data. Due to the limited duration of the dataset (three years), our conclusions will be based on this relatively short time span. Nevertheless, analyzing this timeframe can still provide valuable information about the behavior of temperature, humidity, pressure, wind velocity, and wind direction.

We present three figures to illustrate the daily and yearly evolution of temperature. The first figure 3.11 depicts the daily temperature variations over the recorded period, showing separate lines for each day. This visualization allows us to observe the fluctuations and patterns within each day.

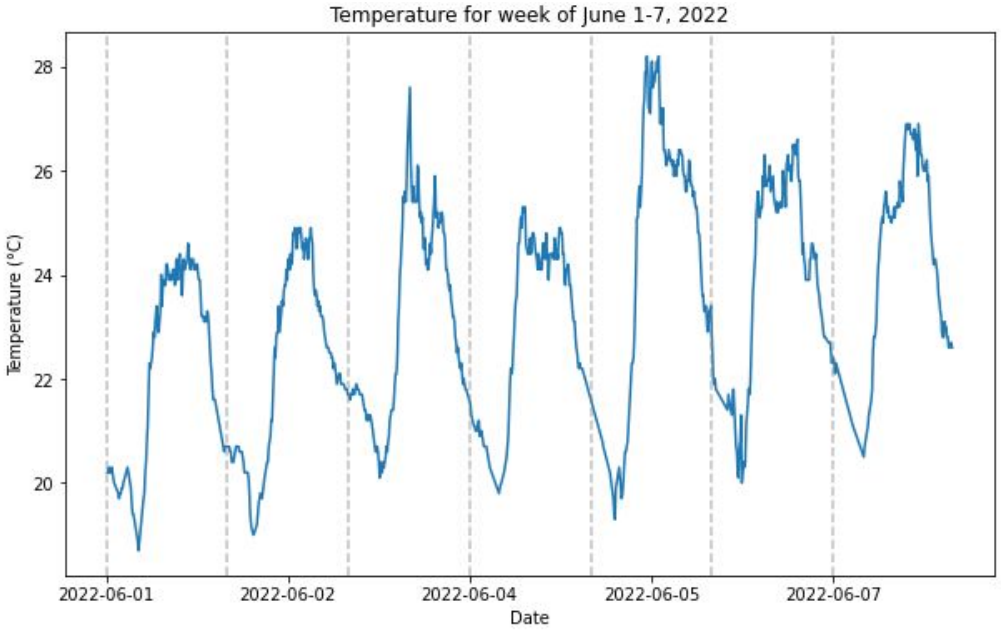


Figure 3.11: Daily evolution of temperature during a week.

The second figure 3.12 shows the yearly evolution of temperature, highlighting the seasonal variations. The lines representing different years enable us to compare temperature trends across multiple seasons and identify any recurring patterns.

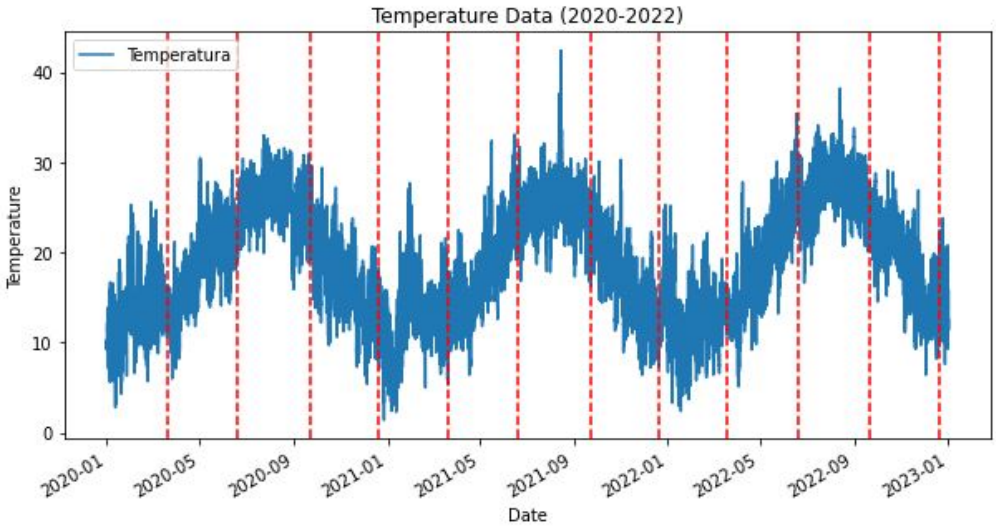


Figure 3.12: Yearly evolution of temperature.

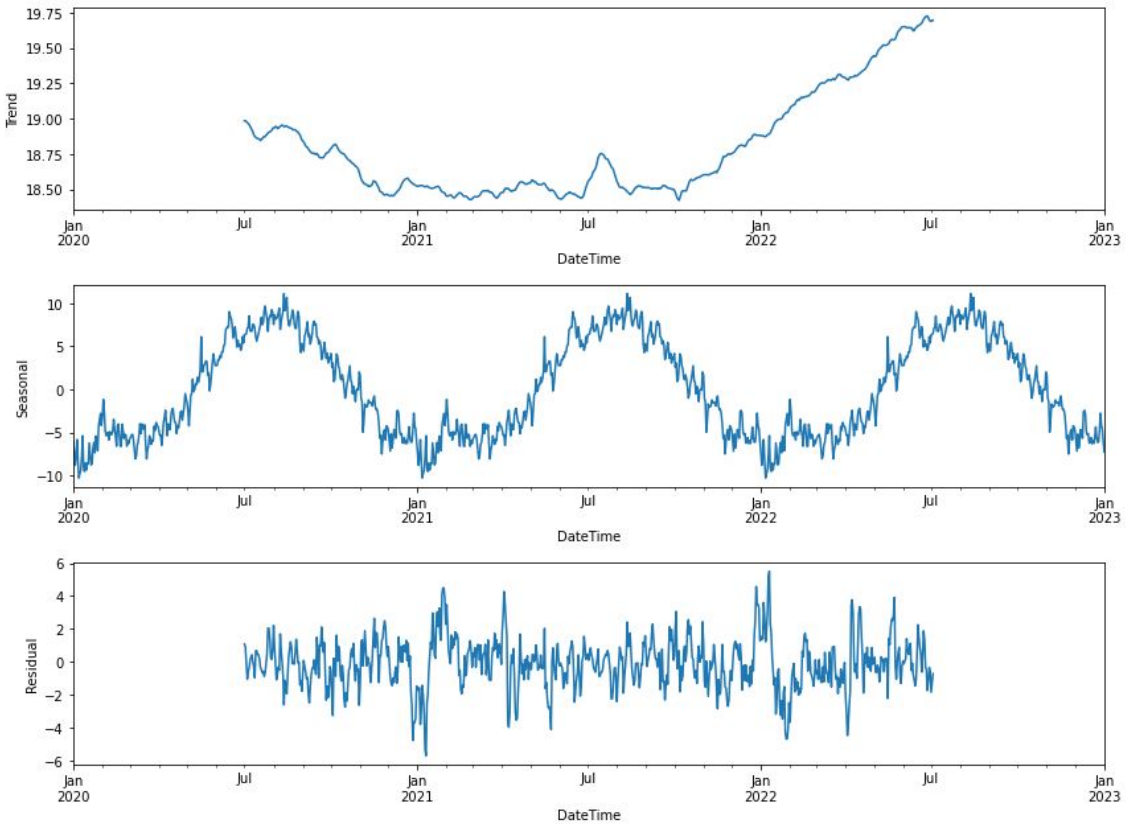


Figure 3.13: Trends of temperature.

Lastly, we present a figure 3.13 that showcases the overall trend of temperature over the recorded period. Based on our analysis, we observe a tendency of temperature to increase

over time, indicating a potential upward trend. While we acknowledge that three years is a relatively short time frame, this observation suggests the need for further investigation into potential climate change effects or other factors contributing to temperature changes.

By examining these tendencies over time, we can gain insights into the variations, patterns, and potential trends of temperature. However, it is important to note that our conclusions are based on a limited dataset, and further analysis with a longer time span would be necessary to draw more robust and definitive conclusions.

3.2.6 Data Understanding

By performing a comprehensive analysis of the data collected from the weather station, including examining the distributions, correlations, variable comparisons, and tendencies, we gain valuable insights into the weather conditions observed at the location. This analysis forms the foundation for further investigations and can contribute to better understanding and predicting local weather patterns and phenomena. Moreover, we realise that all variables might be included as features in the following model for time series forecasting.

3.2.7 Model Architecture

For the prediction of time series variables, we have developed a deep learning model that incorporates a combination of convolutional and dense layers. This architectural choice is motivated by the convolutional layers' ability to capture spatial patterns in the input data, which is particularly advantageous for analyzing time series data.

To optimize the model's hyperparameters, we utilized the Optuna[3] library. Optuna provides a framework for automating hyperparameter optimization through efficient search algorithms. By defining a search space that includes hyperparameters such as the number of convolutional and dense layers, filters, kernel sizes, dropout rates, and learning rates, it performs a systematic search to identify the configuration that minimizes the loss function.

To illustrate the implementation of Optuna for hyperparameter optimization, consider the following code snippet:

```
def objective(trial):
    n_layers_conv = trial.suggest_int('n_layers_conv', 3, 5)
    n_layers_dense = trial.suggest_int('n_layers_dense', 3, 5)
    batch_size = trial.suggest_categorical('batch_size', [16])
    optimizer = trial.suggest_categorical('optimizer', ['adam'])
    learning_rate = trial.suggest_loguniform('learning_rate',
```

[3] https://optuna.org/

```
                    1e-5, 1e-3)
    dropout_rate = trial.suggest_uniform('dropout_rate', 0.0, 0.5)
    n_filters = trial.suggest_int('n_filters', 64, 128, step=32)
    kernel_size = trial.suggest_int('kernel_size', 3, 5)
    n_neurons_dense = trial.suggest_int('n_neurons_dense', 128,
                        256, step=64)

    model = create_model(n_layers_conv, n_layers_dense, batch_size,
                        optimizer, learning_rate, dropout_rate,
                        n_filters, kernel_size, n_neurons_dense)

    history = model.fit(X_train, y_train, validation_split=0.2,
                        batch_size=batch_size, epochs=10, verbose=1)

    return history.history['val_loss'][-1]

study = optuna.create_study(direction='minimize')
study.optimize(objective, n_trials=10)
```

In this example, the objective function defines the search space for the hyperparameters and trains the model using the suggested hyperparameters. The Optuna library then performs the hyperparameter search for a specified number of trials, and the best hyperparameters and corresponding loss value are obtained from the study object.

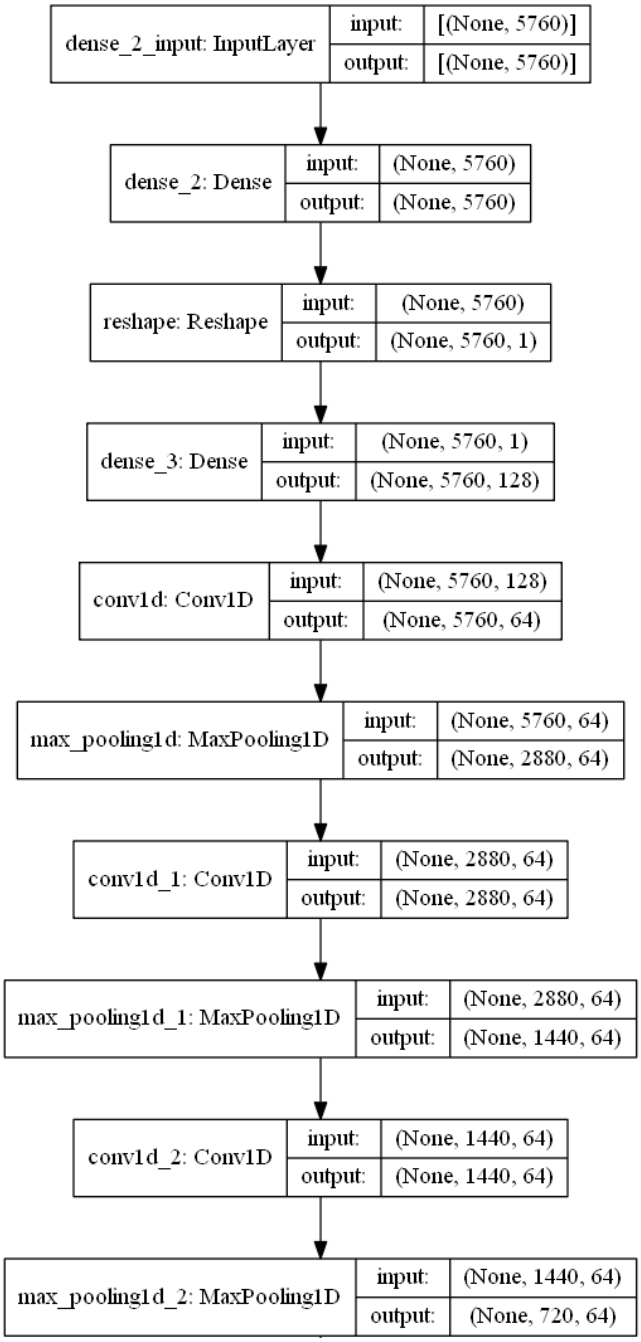
By employing Optuna, we can efficiently explore the hyperparameter space and select the optimal configuration for our model. This approach helps us fine-tune the model's performance and improve its accuracy in predicting the time series variables. The final architecture is shown in Figure 3.14. The different layers election is justified as follow:

- Convolutional Layers. The model incorporates multiple convolutional layers to extract relevant features from the input sequence. These layers utilize filters with different kernel sizes to scan the input data and capture local patterns and dependencies. By applying the ReLU activation function and using appropriate padding, the convolutional layers enhance the model's ability to identify important temporal patterns in the time series data.

- Max Pooling Layers Layers. Max pooling layers are inserted after each convolutional layer to downsample the feature maps and reduce the spatial dimensions. This helps to reduce

the computational complexity and extract the most salient features while preserving important information.

- Dense Layers. The model incorporates several dense layers to further process the extracted features and capture higher-level representations. The dense layers have a large number of units, promoting the model's capacity to learn complex relationships in the data. The ReLU activation function is used in these layers to introduce non-linearity and enhance the model's ability to model intricate patterns.



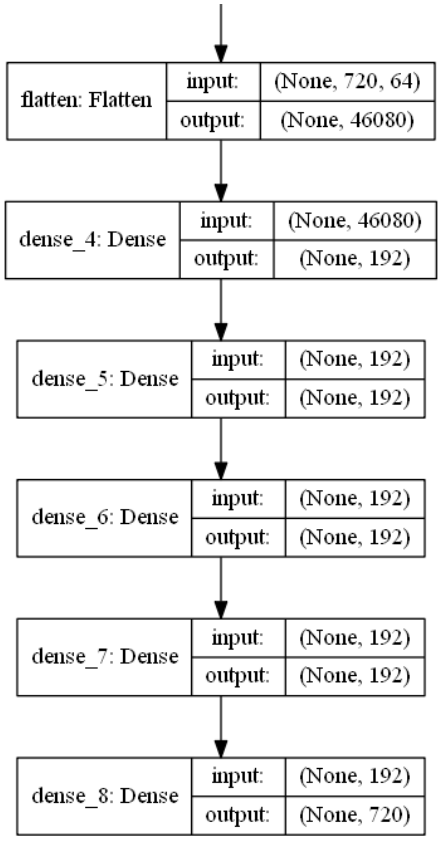


Figure 3.14: Architecture of the time series prediction model

To train the model, we utilize the Huber[4] loss function instead of traditional loss functions like mean squared error. The choice of Huber loss in this context is particularly interesting due to the nature of the data and potential outliers that can occur. Weather variables such as temperature, humidity, and pressure are subject to various factors and conditions that can result in extreme values or occasional anomalies in the dataset.

Huber loss is known for its robustness to outliers, meaning it provides a balanced approach to handling data points that deviate significantly from the expected values. Unlike traditional loss functions like mean squared error (MSE), which heavily penalize outliers, Huber loss combines the benefits of mean absolute error (MAE) and MSE by providing a smooth transition between the two.

By incorporating Huber loss, the model can effectively handle and mitigate the impact of outliers in the training process. This allows the model to learn from both normal and extreme weather conditions, leading to more accurate predictions and improved overall performance.

During the training process, we employ early stopping as a technique to prevent overfitting and improve generalization. Early stopping monitors the validation loss during training and

[4] https://www.tensorflow.org/api_docs/python/tf/keras/losses/Huber

stops the training process if the validation loss fails to improve over a specified number of epochs. This helps us find the optimal balance between model complexity and generalization ability, reducing the risk of overfitting the training data.

The model is trained using the Adam optimizer with a learning rate of 0.0001. Adam is an adaptive learning rate optimization algorithm that adjusts the learning rate based on the gradient of the loss function. This allows the model to converge faster and potentially achieve good results.

To monitor the training progress and assess the model's performance, we capture the learning curve through the training history. The history object stores information about the loss and metrics computed during training and validation for each epoch. Since the cross validation error is higher than the training error, we proof that there is no overfitting(3.15).

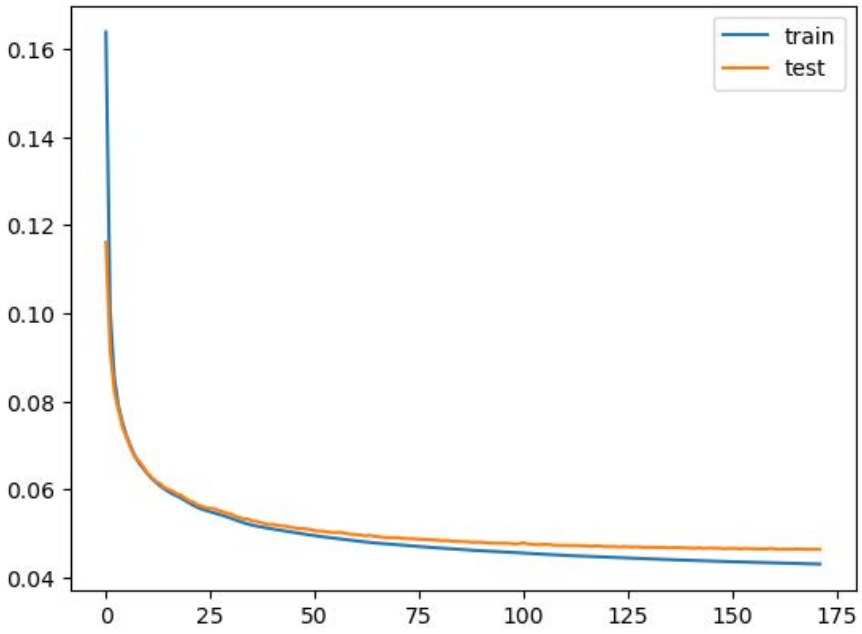


Figure 3.15: Loss time series model

3.2.8 Choosing the Size of the Output Window

To determine the size of the input window for our time series prediction model, we conducted experiments with various input window sizes. The output window was set to one day which consists of 144 timesteps. We trained and evaluated the model using different input window sizes to assess their impact on the model's performance.

Table 3.2.8 shows the results of these experiments, including the validation loss, the number of epochs until early stopping (ES), the average time per epoch, the total training time, and

the number of parameters in the model.

To ensure the reliability of the experiments, multiple runs were conducted, and the median results are reported in the table. The experiments consistently demonstrated deterministic behavior, with the training loss showing a decreasing trend throughout the training process.

Days	Input Window	Val_Loss	Epochs (ES)	Time / Epoch (s)	Total Time (s)	Params
1	144	0.0554	284	24	6816	1957728
2	288	0.0503	306	32	9792	4619568
3	432	0.0492	237	27	6399	8318208
4	576	0.0477	231	49	11319	13053648
5	720	0.0470	245	58	14210	18825888
6	864	0.0472	183	67	12261	25634928
7	1008	0.0462	198	75	14850	33480768
8	1152	0.0460	234	87	20358	42363408

Table 3.3: Results of experiments with different input window sizes

As shown in the table, varying the input window size had an impact on the model's performance. Smaller input windows, such as 144 and 288 timesteps, resulted in higher validation losses and required more epochs to converge. On the other hand, larger input windows, such as 1008 and 1152 timesteps, achieved lower validation losses and required fewer epochs to converge.

Among the various input window sizes tested, an input window size of 1008 stands out as the most favorable choice. This decision is based on the following justifications:

- Validation Loss: The experiment with an input window size of 1008 achieved a validation loss of 0.0462, which is almost the lowest compared to the other sizes. A lower validation loss indicates that the model's predictions were closer to the actual values, demonstrating its superior performance in capturing the underlying patterns in the data.

Although an input window size of 1008 is computationally and time demanding, it provides the advantage of achieving the lowest validation loss. The model tends to reach a point of convergence with this input window size, indicating that it captures the important patterns and dependencies in the data. While training time may be longer, the improved prediction accuracy justifies the choice of this input window size.

Additionally, we plotted the validation loss against the number of parameters in the model, as shown in Figure 3.16. This plot provides further insights into the relationship between the model's complexity (measured by the number of parameters) and its performance (measured by the validation loss).

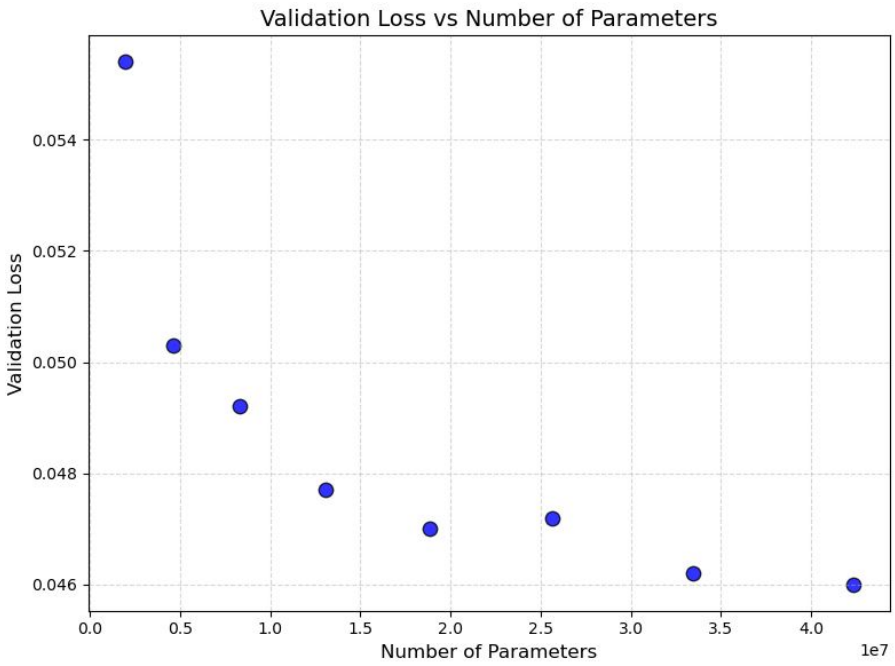


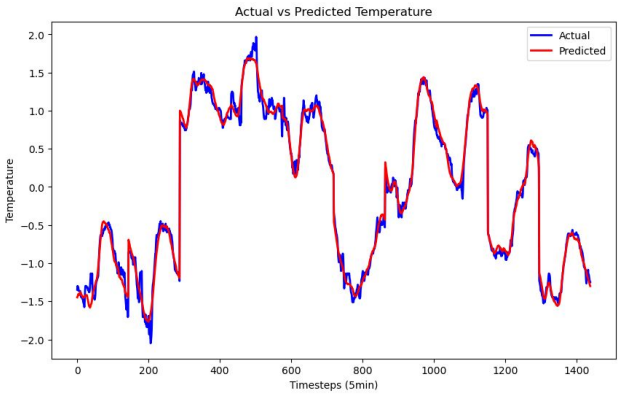
Figure 3.16: Validation Loss vs Number of Parameters

The scatter plot in Figure 3.16 illustrates that as the number of parameters increases, the validation loss tends to decrease. This suggests that a more complex model with a larger number of parameters has the potential to capture finer-grained patterns in the data, leading to improved prediction accuracy. However, it's important to consider the trade-off between model complexity and computational resources, as larger models require more computational power and longer training times.

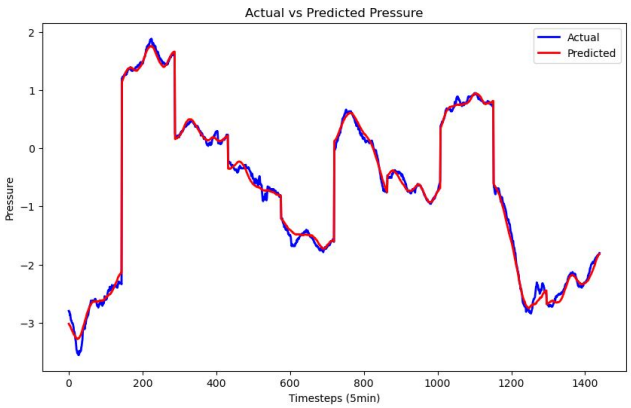
In conclusion, our experiments demonstrate the impact of input window size on the performance of our time series prediction model. Based on the results, an input window size of 1008 is recommended for achieving the lowest validation loss. Furthermore, the plot of validation loss against the number of parameters provides insights into the relationship between model complexity and performance, highlighting the potential benefits of larger models in capturing intricate patterns in the data.

3.3 Prediction Results

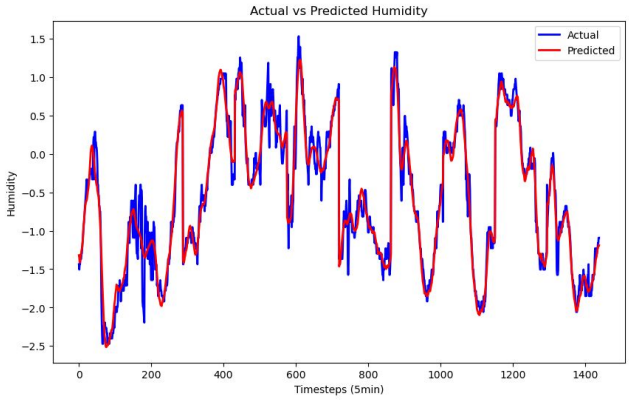
In this section, we present the prediction results for each variable using the trained model. For each variable, we provide a plot comparing the actual values with the predicted values, along with the corresponding MSE metric to assess the accuracy of the predictions.



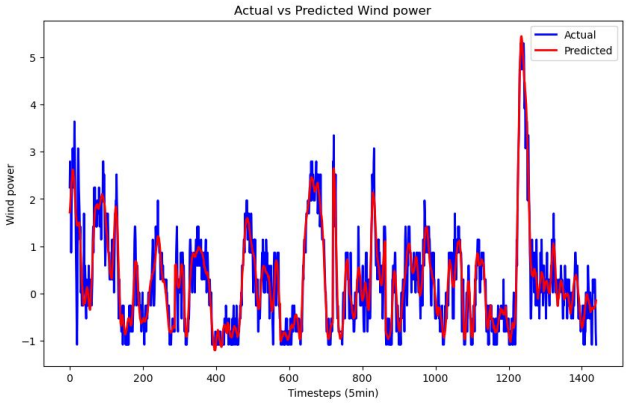
(a) Temperature



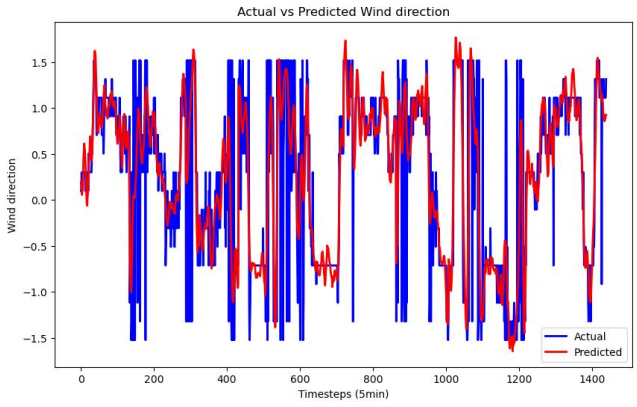
(b) Pressure



(c) Humidity



(d) Wind velocity



(e) Wind Direction

Figure 3.17: Performance evaluation of weather variables

The overall evaluation error for the five variables (temperature, humidity, pressure, wind velocity, and wind direction) is 0.0641. It represents the MSE calculated across all the predicted values for these variables.

It is important to note that our predictions are focused on punctual forecasting rather

than long-term trend forecasting. Nevertheless, to provide a comprehensive visualization of the model's performance, we will be plotting a concatenation of predicted and actual values for each variable. We will also calculate the MSE for each variable to assess the accuracy of the predictions.

The figures presented in Figure 3.17 showcase the predicted and actual values for the five weather variables, accompanied by their corresponding MSE values (3.3). These visualizations and MSE values provide valuable insights into the overall performance of the model in predicting each variable.

Upon examining the plots, it is evident that the model's predictions for some variables are quite accurate, while others exhibit higher deviations. Specifically, the predictions for temperature, humidity, and pressure (Figures 3.17(a), 3.17(c), and 3.17(b), respectively) closely align with the actual values. This suggests that the model successfully captures the underlying patterns and dynamics associated with these variables.

Conversely, the predictions for wind velocity and wind direction (Figures 3.17(d) and 3.17(e), respectively) demonstrate comparatively larger deviations from the actual values. This discrepancy is understandable, as wind velocity and direction are inherently more challenging to forecast accurately due to their complex and dynamic nature. Despite these challenges, the model still produces predictions that provide valuable insights, albeit with slightly higher errors.

Quantitatively, the MSE values further validate the observations made from the plots. As summarized in Table 3.3, the MSE values for temperature, humidity, and pressure are relatively low, indicating a high level of accuracy in predicting these variables. On the other hand, the MSE values for wind velocity and wind direction are higher, consistent with the visual observations.

Variable	MSE
Temperature	0.0096
Humidity	0.0064
Pressure	0.0348
Wind Velocity	0.2639
Wind Direction	0.1530

Table 3.4: Summary of MSE for each variable

In conclusion, the presented figures and MSE values collectively demonstrate the model's overall performance in predicting the weather variables. The model exhibits accurate predictions for temperature, humidity, and pressure, while the predictions for wind velocity and wind direction, although relatively less accurate, still provide valuable insights. These findings highlight the inherent challenges associated with forecasting wind-related variables and emphasize

the model's ability to capture the underlying patterns in the majority of the weather data.

3.4 Validation

In this section, we validate the hypothesis that the trained model for a specific weather station can be applied to every point in the grid built earlier. The validation process involves the following steps: interpolation, model feeding with interpolation, dataset creation using the model of interpolation, and final model inference on the dataset.

First, as done previously with the rain, we conduct the grid creation across the city. Next, we perform interpolation to estimate the weather values at each dot location. We use the best interpolation method with less loss achieved before which is RBF to interpolate the weather data from the surrounding weather stations. This step helps us obtain a dataset with the estimated values where measurements are not available.

The interpolated data from the previous step is transformed and then the model is fed. The model takes the dataset values as input and predicts the time series of various variables, including humidity, temperature, pressure, wind velocity, and wind direction, for a specific time period.

Finally, we apply the trained model to the dataset created using the interpolated variables. The model performs inference on the dataset and generates predictions for the variables of interest. These predictions provide insights into the behavior and patterns of the selected variables across different locations in the city.

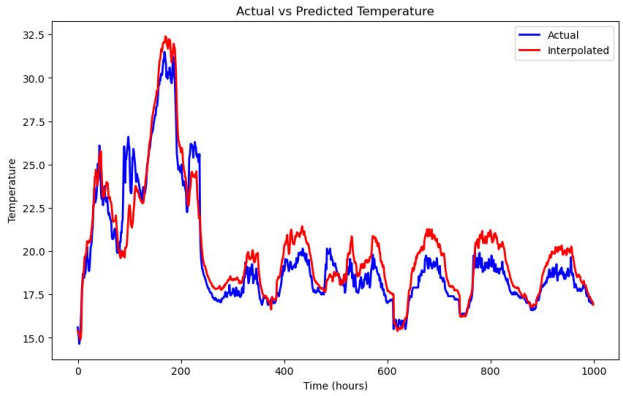
By following this validation process, we can assess the applicability of the trained model to every point in the grid and evaluate the accuracy of the predictions for the selected variables.

In the next subsections, we present the results of the validation process, including the predicted values and their comparison with actual measurements, to evaluate the effectiveness of the model in capturing the spatiotemporal dynamics of the weather variables across the city.

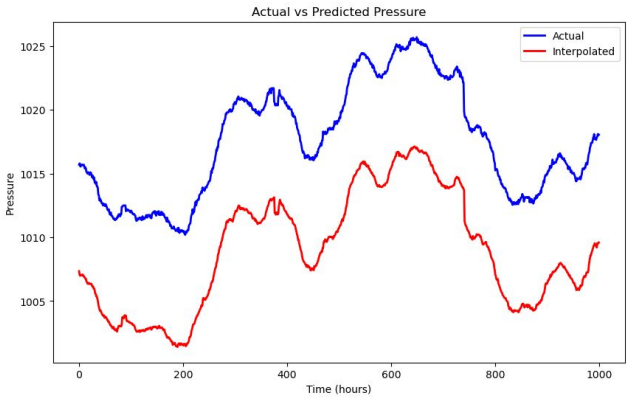
3.4.1 Interpolation Validation

The interpolation validation results indicate that the method performs well in predicting the majority of the weather variables. The plots in Figure 3.18 demonstrate the predicted values compared to the actual values for each weather variable: temperature, pressure, humidity, wind velocity, and wind direction.

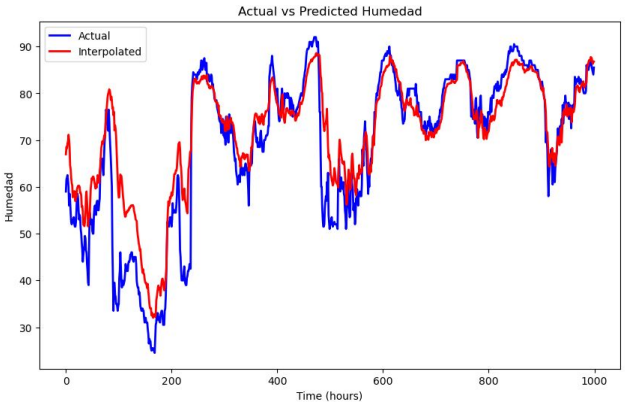
However, it is worth noting that the pressure variable exhibits a vertical bias in the predictions (Figure 3.18(b)). The interpolation method struggles to capture the vertical variations in pressure accurately, resulting in a noticeable deviation between the predicted and actual pressure values.



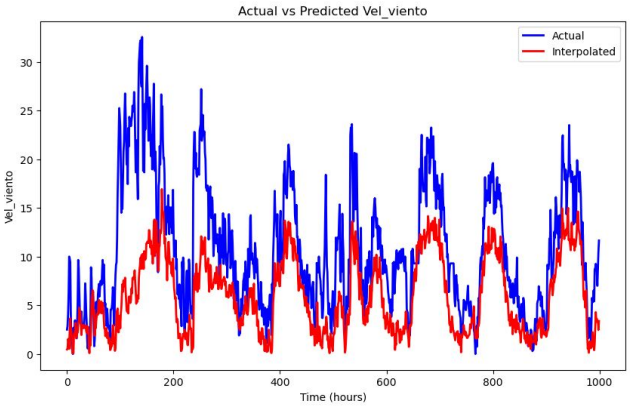
(a) Temperature



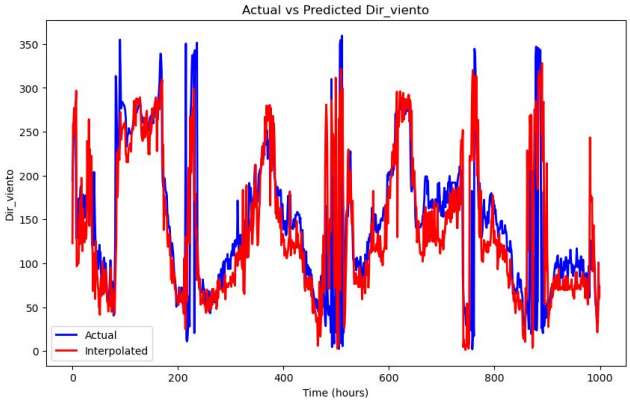
(b) Pressure



(c) Humidity



(d) Wind Velocity



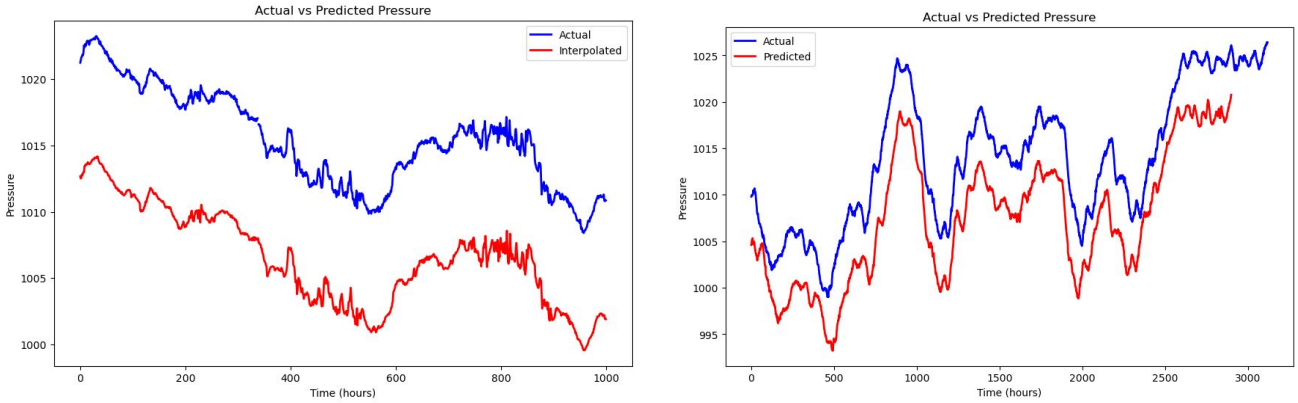
(e) Wind Direction

Figure 3.18: Validation of interpolation in Capitol

The pressure problem in the interpolation arises from the presence of a vertical bias, where the predicted pressure values exhibit systematic deviations from the actual values along the vertical axis. This bias indicates that the RBF method seems to not be suitable for accurate pressure predictions.

Given the vertical bias, it is not appropriate to conduct machine learning-based inference for pressure forecasting at this stage. It is necessary to investigate and address the underlying causes of this bias before relying on the model's predictions for practical applications.

To understand why the vertical bias is occurring, the pressure interpolation was further validated using two additional weather stations: Ribalta and the private source. The pressure interpolation results at these stations are depicted in Figure 3.19(a) and Figure 3.19(b), respectively.



(a) Ribalta

(b) Private source

Figure 3.19: Validation of interpolation in Ribalta and private source

By evaluating the pressure interpolation across multiple weather stations, we can gain insights into the consistency of the vertical bias. If all three stations exhibit a similar bias, it suggests that the issue is not specific to a particular location or dataset. This information is valuable for identifying the potential causes of the bias and developing strategies to mitigate it.

The vertical bias observed in the pressure interpolation can be attributed to possible sensor errors and spatial variability. These factors can introduce inaccuracies in the recorded pressure values and affect the model's ability to make accurate predictions.

Sensor errors refer to potential inaccuracies or inconsistencies in the pressure sensors used at the weather stations. These errors can arise from calibration issues, drift in sensor performance over time, or other technical limitations. Such errors can propagate into the collected data and lead to biased pressure readings.

Spatial variability, on the other hand, pertains to the natural variations in atmospheric pressure across different locations. Factors such as local weather patterns, geographical features, and atmospheric conditions can influence the pressure distribution. If the model fails to adequately capture these spatial patterns, it may struggle to produce accurate predictions,

resulting in the observed vertical bias.

To mitigate the vertical bias in pressure interpolation, one straightforward solution could be to introduce a constant value to the calculated pressure. This adjustment could be determined based on the difference between their maximum pressure values recorded, denoted as “a” and “b”. By calculating the difference (max(a) - max(b)), we can estimate the magnitude of the bias and add a constant value to the predicted pressures to compensate for it. However, it’s important to note that this solution is not perfect and should be considered as a temporary measure due to the lack of time for more in-depth investigation and analysis and it is out of the scope of this work.

While this approach may help alleviate the bias to some extent, it is crucial to emphasize that it is not a definitive or ideal solution. The underlying causes of the bias, such as sensor errors and spatial variability, should be thoroughly investigated to understand the root sources of the problem. This investigation may involve examining the sensor calibration processes, analyzing the data collection methods, and considering advanced techniques for spatial modeling and interpolation.

After applying this adjustment, we have re-evaluated the pressure interpolation using the Capitol weather station as a validation dataset. The results, shown in Figure 3.20, demonstrate a significant improvement in the fit, with a vertical deviation of 8.5 milibars. This indicates that the constant value adjustment has effectively reduced the vertical bias in the pressure predictions.

With this improved pressure interpolation, we can now proceed with the inference of the machine learning model for time series prediction using the dataset created from the interpolation. The adjusted pressure values contribute to more accurate input data, enhancing the model’s ability to capture the underlying patterns and make reliable predictions.

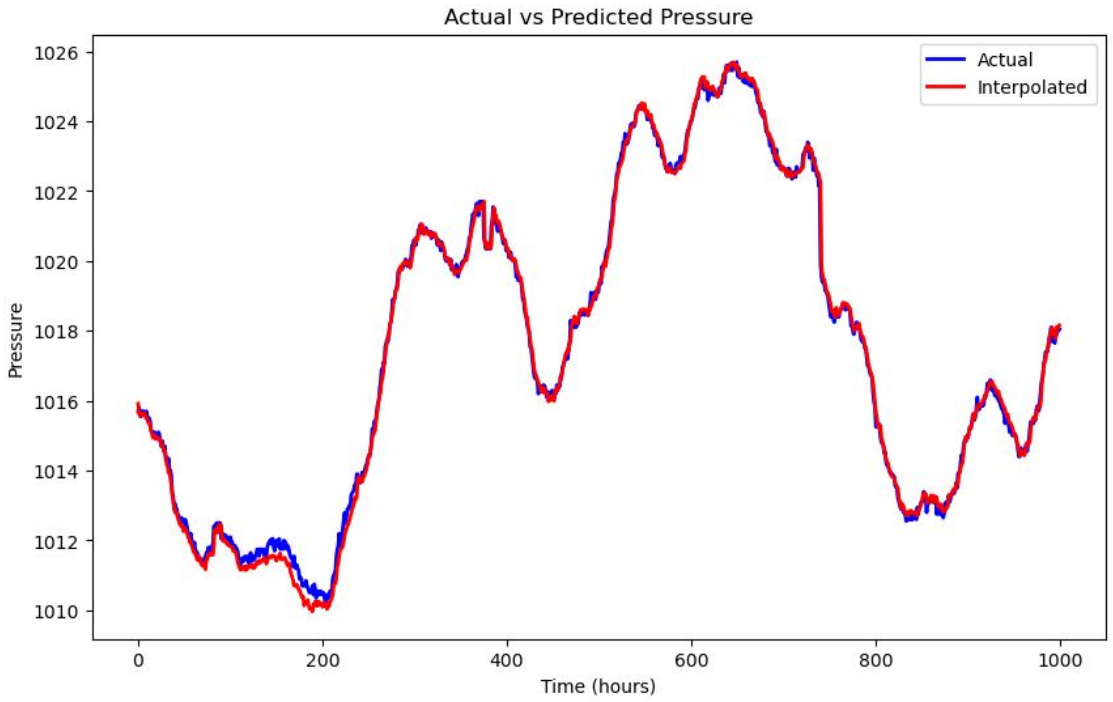
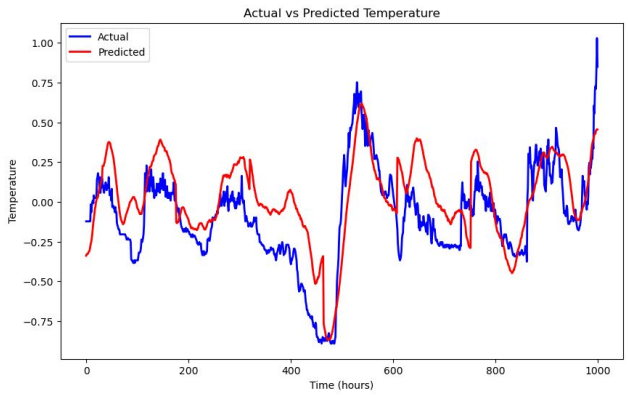


Figure 3.20: Pressure Interpolation with Constant Value Adjustment (Capitol Weather Station)

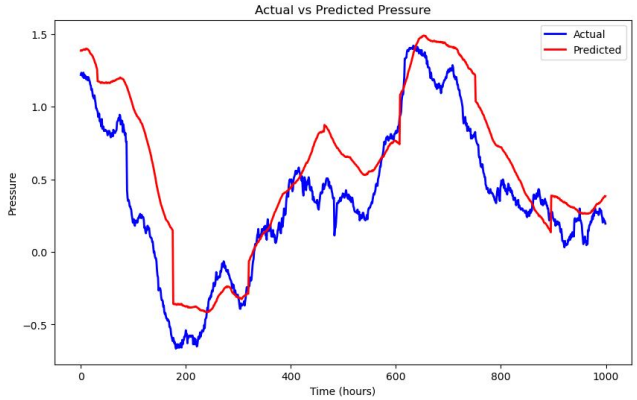
3.4.2 Inference of the Weather Analysis Model

Figure 3.21 depicts the predicted and actual values for the five weather variables: temperature, pressure, humidity, wind velocity, and wind direction. The model's predictions show good overall precision, with notable accuracy in pressure, temperature, and humidity. However, wind velocity and wind direction exhibit higher MSE values, which is expected as they are inherently more challenging to forecast due to their unpredictable nature.

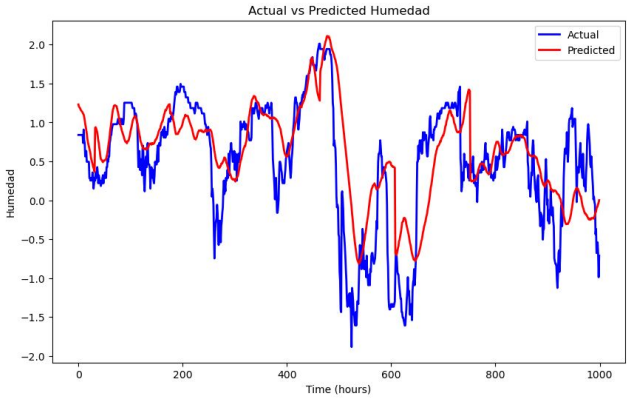
In the evaluation of the time series forecasting model, it is important to note that the figures depicting the predicted and actual values for the weather variables are plotted with normalized values.



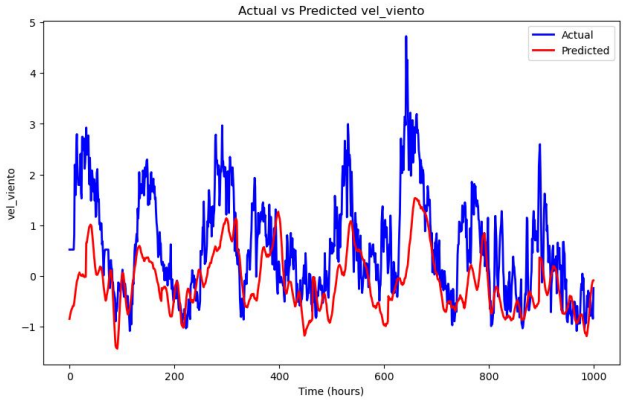
(a) Temperature



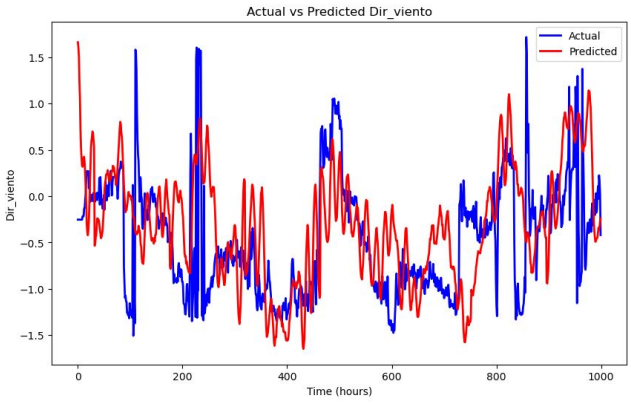
(b) Pressure



(c) Humidity



(d) Wind Velocity



(e) Wind Direction

Figure 3.21: Final performance evaluation of weather variables

Table 3.4.2 presents the MSE values for each weather variable, providing a quantitative assessment of the model's accuracy in predicting the respective variables. The lowest MSE values are observed for pressure, temperature, and humidity, indicating better prediction performance for these variables. The higher MSE values for wind velocity and wind direction align with

their inherent difficulty in forecasting due to their volatile and less predictable nature.

Variable	MSE
Temperature	0.052
Humidity	0.09
Pressure	0.37
Wind Velocity	1.08
Wind Direction	0.45

Table 3.5: Summary of MSE for each variable

3.4.3 Final Thoughts

In conclusion, the evaluation of the time series forecasting model reveals promising results for predicting various weather variables. The model demonstrates good performance in forecasting temperature and humidity, which are relatively predictable variables. The predictions for pressure, while improved with the constant value adjustment based on the difference between reference weather stations, still exhibit a noticeable vertical bias.

It is crucial to address the issue of pressure bias in the interpolation process, as it may strongly affect the accuracy of weather predictions in certain scenarios. Although the constant value adjustment provides a straightforward solution, further investigation is needed to understand the underlying causes of the bias and explore more precise methods to correct it. The source of the bias could potentially be attributed to sensor errors, spatial variability, or other factors that warrant closer examination.

Despite the challenges posed by the pressure bias, the model's performance in predicting other weather variables, such as temperature and humidity, remains satisfactory. These variables exhibit lower MSE values, indicating higher accuracy in their predictions. It is important to note that wind velocity and direction, being inherently more volatile and subject to various atmospheric conditions, naturally pose greater challenges for prediction.

Overall, the time series forecasting model shows promise for inferring future weather conditions. However, addressing the pressure bias and exploring further improvements will be crucial for enhancing the overall accuracy and reliability of the predictions.

Chapter 4

Conclusions

4.1 Future Work

In this section, we outline potential directions for future work that could extend and improve upon the findings and methodologies presented in this study. These ideas aim to address certain limitations and explore new avenues in the field of weather forecasting and analysis.

4.1.1 Testing New Models

While our current model architecture has shown effectiveness in time series forecasting, it would be valuable to test and evaluate new models to further enhance prediction accuracy. The following models could be considered for future experimentation:

XGBoost is a popular ensemble learning algorithm that combines the predictions of multiple weak models, such as decision trees, to create a more accurate and robust model. Exploring the application of XGBoost for time series prediction could provide insights into the performance of gradient boosting techniques and their potential for improving prediction accuracy.

Ensemble of Neural Networks can also be applied to neural networks to enhance prediction performance. By combining the predictions of multiple neural network models, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), we can leverage the diverse capabilities of these models to capture different aspects of weather patterns and improve overall forecasting accuracy.

4.1.2 Transformer Models

Transformer models have emerged as state-of-the-art architectures for various tasks, including natural language processing and computer vision. They have also shown promising results in spatiotemporal data analysis, making them an interesting avenue for weather forecasting. The following transformer-based models could be explored:

The approach proposed by Chattopadhyay et al. (11) integrates data assimilation techniques with a deep spatial-transformer-based U-NET. By assimilating real-time observational data into the forecasting model, this approach aims to improve the accuracy and reliability of predictions. Exploring the integration of this model into our framework can provide valuable insights into the benefits of combining physics-inspired data-driven approaches with our existing models.

The novel Transformer network with shifted window cross-attention as proposed by Bojesomo et al. (12) is specifically designed for spatiotemporal weather forecasting. This model leverages the shifted window cross-attention mechanism to effectively capture the spatial and temporal patterns in weather data. Integrating this model into our framework can enhance the ability to capture complex spatiotemporal relationships and improve the accuracy of weather predictions.

By exploring these new models and incorporating them into our forecasting framework, we can continuously improve the accuracy and reliability of weather predictions, paving the way for more advanced and robust forecasting systems.

4.1.3 Distributed Training

In this study, we focused on developing a relatively small and simple model for time series prediction. However, it is recommended to explore the potential of using larger models and conducting distributed training to fully leverage the capacity of available GPUs. Indeed, if all the weather stations are used to feed the time series forecasting model instead of only one source, it will be a must to look forward distributed training approaches.

It is observed that utilizing a single GPU significantly speeds up the training process compared to using a CPU. The training time using a CPU is approximately 30 times slower than training with a single GPU. Therefore, for future work involving larger models, it is crucial to employ GPU-based training to reduce the training time.

To further investigate the performance differences between GPU and CPU training and to fully exploit the capabilities of high-performance computing, future work can follow the workflow outlined in the paper by Gonzalez et al. (13). This will enable a comprehensive evaluation of distributed training strategies and performance optimization techniques for time series prediction models.

4.1.4 Incorporating more External Data Sources

To further enhance the accuracy and predictive capabilities of our models, incorporating more external data sources can be valuable. Weather patterns and their impacts are influenced by various factors, such as satellite imagery, geographical features, and climate indices. By integrating these data sources into our models, we can capture additional information and improve the quality of weather predictions.

4.1.5 Integration of Uncertainty Estimation

Accounting for uncertainty in weather analysis is crucial for decision-making processes. Incorporating techniques such as Bayesian deep learning or Monte Carlo dropout into our models can enable us to estimate the uncertainty associated with our predictions. This information can assist users in understanding the reliability of the forecasts and guide them in making informed decisions.

4.1.6 Exploring Advanced Interpolation Techniques

Although we have employed interpolation techniques to estimate weather variables across the entire grid, there is scope for exploring advanced interpolation methods. For example, machine learning-based interpolation models, such as Gaussian processes or neural network-based interpolators, could be investigated to improve the accuracy and efficiency of the interpolation process.

4.1.7 Integration of Spatial Dependencies

Weather patterns often exhibit spatial dependencies, where the conditions at one location are influenced by nearby locations. Incorporating spatial dependency models, such as spatial autoregressive models or convolutional neural networks, can capture these dependencies and lead to more accurate and coherent forecasts across the entire region of interest.

4.1.8 Integration of Real-Time Data Streams

Real-time weather data streams provide up-to-date information that can enhance the timeliness and accuracy of forecasts. Integrating live data feeds from weather stations, remote sensing platforms, and other sources can enable continuous model updating and improve the responsiveness of the forecasting system to rapidly changing weather conditions.

These future research directions hold the potential to advance the field of weather analysis, and we encourage further investigation to explore these avenues and uncover new insights and techniques for improved weather predictions and decision support systems.

4.1.9 Generalization to Different Weather Regions

While our study focused on the specific weather conditions of the Mediterranean coast, it would be valuable to test and evaluate the proposed methods in other regions with distinct weather patterns and characteristics. Different climatic zones, such as tropical, arid, or polar regions, present unique challenges and variations in weather phenomena. By applying our methods to these diverse regions, we can assess their generalizability and adaptability across different weather conditions.

Testing the models and techniques in various regions will provide insights into their performance and robustness in capturing the specific dynamics and complexities of different climate systems. This will allow us to understand the strengths and limitations of the proposed approaches in different geographical contexts and refine them accordingly. Furthermore, it will contribute to building a more comprehensive and versatile framework for weather forecasting and analysis that can be applied globally.

Conducting experiments and evaluations in different regions should consider the availability of data sources, including weather stations, satellite observations, and other relevant datasets specific to the respective regions. Additionally, taking into account local climatic factors, topographical characteristics, and meteorological phenomena will be essential for accurate model training and validation.

By extending the study to encompass a broader range of weather regions, we can gain a deeper understanding of the effectiveness and applicability of the proposed methods across diverse climatic conditions. This will not only enhance the field of weather forecasting but also provide valuable insights for localized forecasting systems and decision-making processes in various parts of the world.

4.2 Conclusions

In this study, we proposed a comprehensive framework for weather analysis and time series forecasting that integrates multiple techniques, including dot rain coverage, interpolation, and machine learning models. We conducted an in-depth analysis of rainfall data from weather stations and developed models to predict rainfall patterns and other weather variables.

Our findings highlight the effectiveness of dot rain coverage in capturing the spatial distribution of weather variables across a city. By strategically placing rain dots, we were able to

obtain reliable and representative data points for interpolation and model training. This approach proved to be a valuable step in addressing the issue of limited and unevenly distributed weather stations.

The interpolation techniques employed in this study, namely radial basis function (RBF) and nearest neighbor, effectively filled in the gaps between weather stations and provided interpolated rainfall data for the entire city grid. These interpolated data points were then utilized as inputs for our machine learning models, enabling faster inference compared to direct interpolation during real-time predictions.

We developed and fine-tuned a deep learning model consisting of convolutional and dense layers to forecast various weather variables, including temperature, humidity, pressure, wind power, and wind direction. Through systematic experimentation and hyperparameter optimization, we identified an optimal architecture that yielded accurate predictions with low validation loss.

Furthermore, our study demonstrated the applicability of the developed models across different weather stations within the city. The models trained on one weather station were able to generate accurate predictions for other locations in the city, highlighting the potential for model generalization and transferability.

Overall, our framework provides a holistic approach to weather forecasting and analysis by addressing data sparsity, leveraging interpolation techniques, and utilizing machine learning models. The results showcase the potential for improved predictions and insights into weather patterns, which can benefit various sectors such as agriculture, transportation, urban planning, and disaster management.

In conclusion, our study contributes to the advancement of weather forecasting techniques, particularly in urban settings with limited weather station coverage. By integrating dot rain coverage, interpolation, and machine learning models, we provide a robust and scalable framework for accurate and localized weather predictions. The results encourage further research and application of these techniques in different geographical regions and weather conditions, paving the way for enhanced weather forecasting capabilities and decision-making processes.

Bibliography

[1] M. Fathi, M. H. Kashani, S. M. Jameii, and E. Mahdipour, "Big Data Analytics in Weather Forecasting: A Systematic Review," *Archives of Computational Methods in Engineering*, vol. 29, pp. 1247–1275, 2021.

[2] Z. Karevan and J. A. K. Suykens, "Transductive LSTM for Time-Series Prediction: An Application to Weather Forecasting," *Neural networks : the official journal of the International Neural Network Society*, vol. 125, pp. 1–9, 2020.

[3] P. R. P. G. Hewage, M. Trovati, E. G. Pereira, and A. Behera, "Deep Learning-Based Effective Fine-Grained Weather Forecasting Model," *Pattern Analysis and Applications*, vol. 24, pp. 343–366, 2020.

[4] P. R. P. G. Hewage, A. Behera, M. Trovati, E. G. Pereira, M. Ghahremani, F. Palmieri, and Y. Liu, "Temporal Convolutional Neural (TCN) Network for an Effective Eeather Forecasting using Time-Series Data from the Local Weather Station," *Soft Computing*, vol. 24, pp. 16 453 – 16 482, 2020.

[5] S. Rasp, P. D. Dueben, S. Scher, J. A. Weyn, S. Mouatadid, and N. Thuerey, "WeatherBench: A Benchmark Data Set for Data-Driven Weather Forecasting," *Journal of Advances in Modeling Earth Systems*, vol. 12, 2020.

[6] G. Pellicone, T. Caloiero, G. D. Modica, and I. Guagliardi, "Application of several Spatial Interpolation Techniques to Monthly Rainfall Data in the Calabria Region (southern Italy)," *International Journal of Climatology*, vol. 38, pp. 3651 – 3666, 2018.

[7] U. Bronowicka-Mielniczuk, J. Mielniczuk, R. Obroślak, and W. Przystupa, "A Comparison of Some Interpolation Techniques for Determining Spatial Distribution of Nitrogen Compounds in Groundwater," *International Journal of Environmental Research*, vol. 13, pp. 679–687, 2019.

[8] A. I. Middya and S. Roy, "Spatial Interpolation Techniques on Participatory Sensing Data," *ACM Trans. Spatial Algorithms Syst.*, vol. 7, pp. 13:1–13:32, 2021.

[9] L. Li, X. Zhou, M. Kalo, and R. Piltner, "Spatiotemporal Interpolation Methods for the Application of Estimating Population Exposure to Fine Particulate Matter in the Contiguous U.S. and a Real-Time Web Application," *International Journal of Environmental Research and Public Health*, vol. 13, 2016.

[10] Q. Hu, Z. Li, L. Wang, Y. Huang, Y. Wang, and L. Li, "Rainfall Spatial Estimations: A Review from Spatial Interpolation to Multi-Source Data Merging," *Water*, 2019.

[11] A. Chattopadhyay, M. Mustafa, P. Hassanzadeh, E. Bach, and K. Kashinath, "Towards Physics-Inspired Data-Driven Weather Forecasting: Integrating Data Assimilation with a Deep Spatial-Transformer-Based U-NET in a Case Study with ERA5," *Geoscientific Model Development*, vol. 15, no. 5, pp. 2221–2237, 2022. [Online]. Available: https://gmd.copernicus.org/articles/15/2221/2022/

[12] A. Bojesomo, H. A. Marzouqi, and P. Liatsis, "A Novel Transformer Network with Shifted Window Cross-Attention for Spatiotemporal Weather Forecasting," 2022.

[13] S. Iserte, A. González-Barberá, P. Barreda, and K. Rojek, "A Study on the Performance of Distributed Training of Data-Driven CFD Simulations," *The International Journal of High Performance Computing Applications*, vol. 0, no. 0, p. 10943420231160557, 0. [Online]. Available: https://doi.org/10.1177/10943420231160557