WeatherSplit

Data Splitting Strategies in Weather Prediction

IN5000: Master Thesis

Bogdan-Alexandru Simion

Delft University of Technology

TU Delft

WeatherSplit

Data Splitting Strategies in Weather Prediction

by

Bogdan-Alexandru Simion

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday, July 25, 2024.

Student Number	5850185
Project Duration:	November 15, 2023 - July 25, 2024
Thesis committee:	Prof. Dr. Ir. D.M.J Tax
	Prof. Dr. Ir. M.M. de Weerdt
	R. Ghorbani

Cover:	Canadarm 2 Robotic Arm Grapples SpaceX Dragon by NASA under CC BY-NC 2.0 (Modified)
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

Some of the contents were enhanced using GPT-4.

TU Delft

Preface

This thesis closes the final chapter of my formal academic studies. These 2 years at TU Delft represented a challenge I was willing to accept when I left Romania and it equipped me with plenty of knowledge, both on the professional and personal part.

I would like to express my first special thanks to my supervisor, Professor Tax, as he guided me thoroughly in my work and gave me precious feedback. I would also like to thank Ramin, my PhD supervisor, for all the support and the pieces of advice he gave me throughout these 9 months.

The second thanks round will go towards my parents, who supported me with everything they could, especially in my hardest times, during my studies. Whenever I needed some assistance, they were there to help me unconditionally.

I want to express my gratitude to Rohan for his guidance on everything related to my stay in the Netherlands. I would like to thank him for all the nice times we had here and I hope that we will only grow from this point.

In the end, I would like to thank my friends from campus: Alex, Chelsea, Bobe, Matteo and Roy for all the fun times we had throughout the Master's, and also my Romanian besties Silvia and Alexandra for all the mutual support we had and still have regarding living in another country.

Bogdan-Alexandru Simion
Delft, July 2024

Summary

Weather forecasting has always been a critical issue across various fields, and traditionally, weather predictions have relied on solving complex atmospheric equations using supercomputers. However, the rise of machine learning has introduced a more efficient method that utilizes historical data to forecast weather patterns. Nonetheless, there is a problem with auto-correlation, where values at a specific time point correlate with those at a previous time. Machine learning models assume independence between data points, which can lead to overfitting if the data is not split properly. This happens because training data might not be independent of test data and models can learn the noise rather than the underlying pattern in data. Also, auto-correlation can apply when having multiple locations with data (named spatial data), because nearby locations share similar characteristics.

This thesis investigates temporal and spatial data splitting strategies to address this challenge to determine which methods provide the most reliable performance estimates. Temporal splitting strategies involve dividing the data using time intervals, while spatial splitting strategies involve dividing based on the geographical location of weather stations. The goal is to identify strategies that minimize bias and variance in the error estimates of weather forecasting models. This will ensure less overfitting and robust and generalisable predictions across different climatic conditions and data regimes.

By systematically evaluating various data splitting strategies, this thesis aims to provide insights into the best approaches for preparing meteorological data for Machine Learning-based weather prediction models, thereby contributing to the advancement of more accurate and efficient weather forecasting techniques. Four different strategies for temporal data splitting were evaluated: random splitting, testing at the end of the dataset, testing in the middle of the dataset, and splitting the test year into multiple parts. Three different strategies for splitting spatial data were evaluated: random splitting, the use of one cluster of cities as the test data with the other clusters as training data, and lastly, choosing only the non-neighbouring clusters as training data. Some of these strategies, such as the random strategy, do not take into account auto-correlation when splitting the data.

Results indicate that the random strategy in both scenarios yields a relatively modest error rate. However, it is advisable to employ a strategy that organizes the test data into a cohesive block and positions it in the centre of the dataset time. Hence, this thesis contributes to the development of more refined weather prediction by shedding light on how data splitting strategies influence outcomes.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
AI	Artificial Intelligence
ANN	Artificial Neural Network
BN	Bayesian Network
CNN	Convolutional Neural Networks
CV	Cross-Validation
DL	Deep Learning
HMM	Hidden Markov Models
LDAPS	Local Data Assimilation and Prediction System
LST	Land Surface Temperature
ML	Machine Learning
MLP	Multi-layer Perceptron
NWP	Numerical Weather Prediction
PDF	Probability Density Function
RMSE	Root Mean Squared Error
SAFE	Split Analysis, Full Estimate
S_A	Spatial Strategy a
S_B	Spatial Strategy b
S_C	Spatial Strategy c
STD	Standard Deviation
T_A	Temporal Strategy a
T_B	Temporal Strategy b
T_C	Temporal Strategy c
T_D	Temporal Strategy d

1

Introduction

For ages, weather forecasting has been crucial for many fields, such as agriculture, sporting events, transport, and commerce. Understanding how weather conditions will develop within a given area is an ongoing concern with growing necessity for worldwide affairs. In recent years, though, the climate crisis has affected prediction accuracy, especially in the warmer climates found in mid-latitudes [21, 31, 34].

Weather predictions are usually made using atmospheric equations, which require complex mathematics (e.g., partial differential equations) and can only be solved using supercomputers, which take several hours to compute and render results [28]. Luckily, machine learning (ML) has proved to be useful, as it can provide a simpler and less computationally expensive solution to this problem to predict the weather on any given day, based on past data [5, 10, 13, 15, 17]; however, the evaluation process employs data splitting into train and test sets. There are various methods for splitting data, for example, time series; careful choice of the test set is important, otherwise, the independence between the training and the test set can no longer be guaranteed.

1.1. Preliminaries

A time series is a sequence of data points collected or recorded at successive points in time. Meteorological data such as temperature, humidity, wind speed, and precipitation are typically collected at regular intervals (e.g., hourly, daily, monthly) [30]. It is important to understand the process of collecting meteorological data: data is collected either by weather stations (so-called, *in situ* data) or by satellites. The advantage of satellite measuring is that they have better spatial coverage than weather stations [4]; on the other hand, weather stations have been the main source for measuring the atmosphere regularly for a long time [32]. Therefore, scientists look at *in situ* data when they want to form a bigger picture of the ongoing events on Earth. However, some elements of the Earth's environment can only be measured by the weather stations, such as monitoring of greenhouse gas, where *in situ* data supplies and confirms satellite data, or oceanographic observation, where *in situ* measurements can be obtained from further depths of the ocean [32].

In statistics, auto-correlation refers to the degree of correlation of the same variables between two successive time intervals. Practically, if a variable has a certain value at a time point, then it is more likely for that variable to have a similar value at the next time point. In ML, the auto-correlation can affect the performance of a certain model in numerous ways. First of all, static ML models assume that the observations are independent and identically distributed. Auto-correlation contravenes this assumption, leading to a case where an ML model is performing better on a dataset, but poorer on another dataset. Secondly, auto-correlation can cause models to overfit the training data - when data points are correlated, the model may learn the 'noise' in the data rather than the underlying pattern. This can result in poor generalization to new, unseen data. These aspects are important to be discussed because the auto-correlation is also present in time series and, especially, weather data. As a result, one should be careful when splitting the data into train, validation and test data, as ML models can

exhibit overfitting for one particular data split, and thus, have a lower error than the true error on training for another split [29]. Splitting data in other ways can make the ML models not overfit on the training data by taking longer testing periods, which makes the auto-correlation fade away. In this thesis, dynamic ML models are not covered, although some of the work done can be found in the Chapter 2.

In ML, bias represents the error between the predicted and the true data. Fundamentally, it is the error introduced by approximating a real-world problem, which may be complex, by a simplified model. Variance, in other terms, measures the spread of the data from the mean. Essentially, variance measures the sensitivity of the model to small fluctuations in the training set. In the end, the ideal scenario for an ML model is to have both a low bias and a low variance. For this thesis, we will examine the bias and variance of different data splitting strategies.

1.2. Data splitting strategies

A data splitting strategy refers to a certain way of dividing the data into train, validation and test sets. Choosing a train-validation-test split can take a lot of time, as there is no fixed guideline, especially when handling time series. To address this, Schultz et al. propose some temporal strategies on how to split data, although the authors did not perform a quantitative analysis [29]. Meteorological data can be split on a temporal basis, as in figure 1.1 (as a time series) or spatially (if more weather stations with known coordinates are taken into account). The most common approach in machine learning for splitting time-series data is to split it yearly, as demonstrated in the following section. Another reason for this decision is that weather patterns are cyclic and most of them occur annually. If a dataset contains less than one year of data, then the cyclic pattern is not complete and the model might predict with some inaccuracies.

1.2.1. Temporal splitting strategies

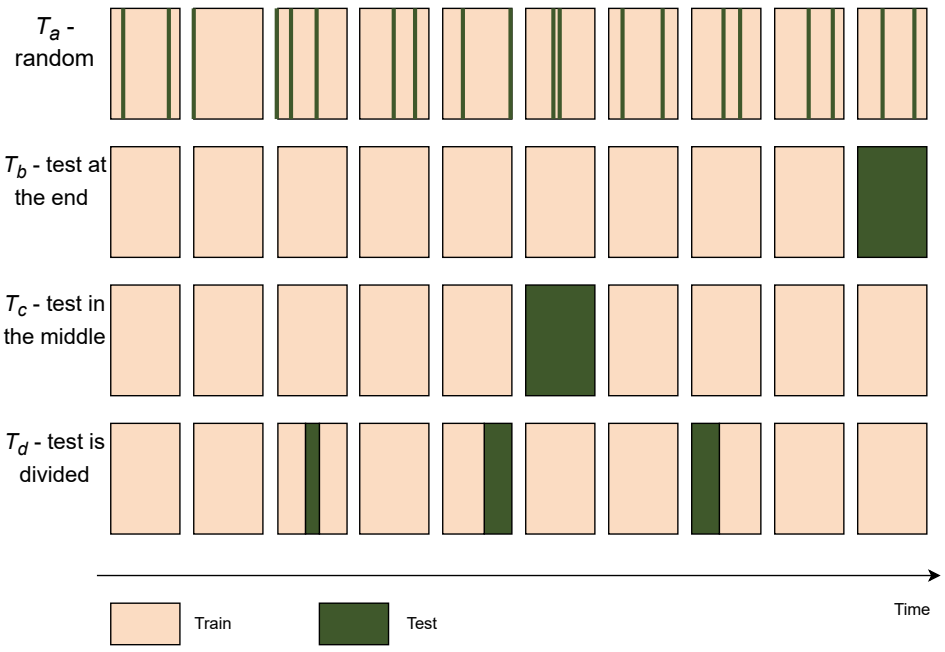


Figure 1.1: Different strategies of temporally splitting a dataset. The blocks represent one year of data and they are separated for clarity. T_a, T_b, T_c and T_d represent the 4 temporal strategies

The first strategy is T_a, which shows a random division without taking into account auto-correlation of the time series (i.e. test set points are randomly scattered within the dataset) [29]. An obvious advantage is the randomness in selecting the train and test data. On the other hand, the test set is comprised of scattered points in the dataset, which renders the test set not independent from the training set. Therefore, by using this strategy, the ML model may overfit and exhibit a high bias, even though the

chosen model is a good fit with regard to the complexity of the data. This strategy is not widely used in Numerical Weather Prediction (NWP).

To tackle this problem, splitting strategy T_b comes into play, where the last period (e.g., one year) is considered as the test set, with the rest being used as the training set [29]. Analysing previous work from the NWP, the T_b strategy was the most commonly used for temporally splitting the data. For instance, Jakaria et al. use 2 months as training data, while the test set comprises 7 days. The forecast is made hourly, by inputting the last 2 days [13]. Holmstrom et al. adopted a 4-fold cross-validation splitting strategy, but with an incremental number of years as training data [10], choosing strategy T_b. Lastly, Lam et al. used strategy T_b as well; training data from 1979 to 2015, 2016 and 2017 were considered validation data and years from 2018 until 2021 were used as test data [17]. One advantage is that this strategy is easily understandable by an inexperienced user and the test data is independent of the training data. However, one disadvantage is that there are not many combinations of data splits, as the test set needs to be at the end of the time frame. Stability and reliability of results need to be taken into account, and thus, strategy T_b might yield high variance if fewer combinations are involved.

The next strategy, T_c, also takes whole periods as test data and randomly places them (e.g., the test set is the first year from the dataset) [29]. Nevertheless, Scher et al. used this strategy, in combination with T_b, using the years between 1985 and 2009, except for 1990 and 2008 (validation data), as training data and years from 2010 to 2016 as testing data [28]. The advantage of this strategy is that one can make more combinations of the data splits to have a mean error that is closer to the true one. A disadvantage, though, is that even though one can make more combinations, some error rates might be beneath the true error, as the model might learn the global trends of the data. For example, let's suppose that the average annual temperatures follow an ascending trend, similar to climate change. As Strategy T_c is placing the test data in the middle of the dataset, the ML model used in training can learn that between two consecutive years, there is a difference in the average temperature. This is not always desirable, because if the model is tested on another dataset that does not take climate change into account, then this strategy might exhibit higher errors.

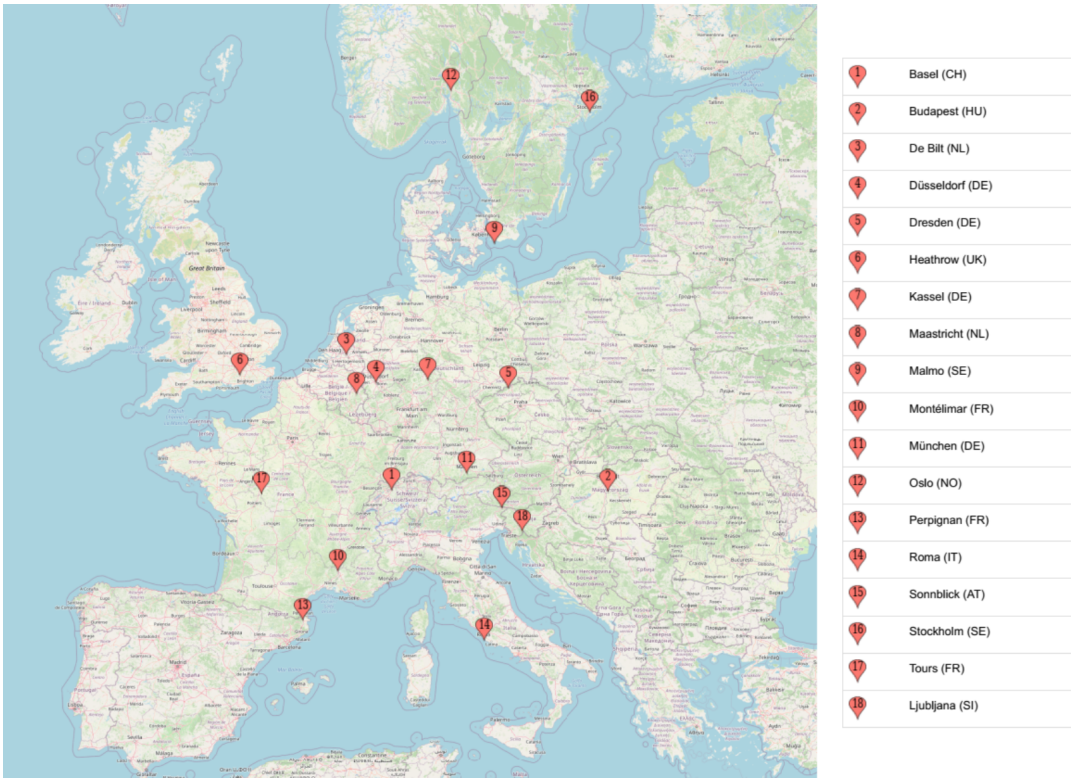


Figure 1.2: Example map of Europe with some weather stations [11]

The last splitting strategy, T_d, divides the validation and test set into smaller, sparser periods throughout

the whole dataset [29]. The advantage of this approach is that it allows for both extrapolation (predicting the future) and interpolation, which theoretically provides the closest estimation of the true error. This can make it the strategy with the lowest bias and variance. Another advantage is that the block of time includes many points, making the correlation fade away. One disadvantage though, is that this splitting strategy is the most difficult for an inexperienced user to implement. This strategy has not yet been studied in NWP.

1.2.2. Spatial splitting strategies

If multiple stations are considered, one can use spatial splitting strategies as well. Figure 1.2 shows an example of weather stations geographically distributed across Europe. There are many ways to split the data in this manner; for example, within a network of weather stations, the station is considered in the test set, while the stations around it are considered as training (and validation) set. Thus, some issues will arise: likewise temporal splitting, how many splitting strategies have a lower error than the true error? One NWP work that has taken spatial splitting into account was performed by Jakaria et al; the authors added neighbouring cities into the training data [13], and their test data was geographically in the centre.

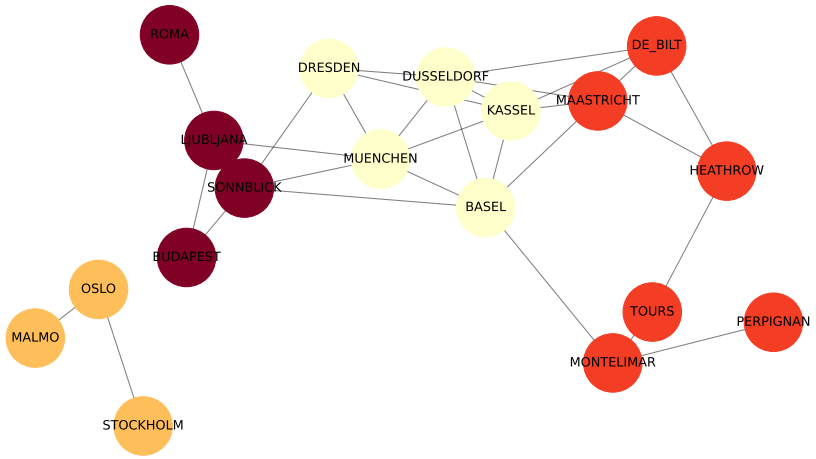


Figure 1.3: Example of clustering the weather stations, where each different node colour represents a different cluster.

Three spatial strategies were assessed: a random strategy, where the data from all stations is combined and random samples are taken for the test set, meanwhile, the rest is considered as training data (depicting the T_a strategy in the temporal setting), marked with S_a; one strategy where the stations are clustered and one is taken as testing, one as validation and the rest are used as the training set (depicts the T_b and T_c strategy in the temporal setting), marked with S_b; and one where one cluster is taken as test set and the non-neighbouring clusters are taken as the training set (no resemblance with the temporal data splitting strategies), marked with S_c. A tentative clustering of some weather stations can be seen in figure 1.3.

1.3. Research questions

This thesis project aims to formulate some theoretical solutions for the following two research questions:

RQ1: "Which temporal data splitting strategy has the lowest bias and lowest variance of the performance in weather forecasting?"

RQ2: "Which spatial data splitting strategy has the lowest bias and lowest variance of the performance in weather forecasting?".

The findings suggest there are no significant differences between the splitting strategies; however, there are some important points. Firstly, strategy T_a (random) yields better results than anticipated and generally does not lead to overfitting. Conversely, strategy T_c is more consistent and therefore, may be more reliable. In contrast, in the spatial scenario, strategy S_a (random) exhibits the smallest disparity between training and test errors. However, it is crucial to consider the research limitations when interpreting the results.

This thesis is organised in the following way: in chapter 2 some related work in the field is presented, to give some more context to the problem. Next, in chapter 3 the setting for the experimental test is discussed, both for temporal and spatial splitting strategies and the data pipeline is explained. The results of the experiments are shown and discussed in chapter 4, and in chapter 5, the limitations of this experiment and some future work are presented, along with some conclusions of this thesis.

2

Related Work

2.1. Cross-validation in time series

This section addresses related work on cross-validation with time series. Although some papers do not cover the NWP ecosystem, they document key insights of processing serial data in domains such as financial as well as recommender systems.

The paper “Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure” by Roberts et al. addresses the challenges of performing cross-validation on ecological data characterized by temporal and spatial dependencies [27]. The authors advocate for the use of block cross-validation, a method that involves partitioning the data into blocks that encapsulate similar behaviours in terms of time or space. This approach acknowledges the inherent structure in ecological data and results in higher error estimates compared to traditional random cross-validation methods [27]. However, these higher errors are indicative of a more accurate representation of the Bayesian error, suggesting that block cross-validation provides a more realistic assessment of model performance when predicting new data or predictor space, particularly in the context of ecological studies where data dependencies are prevalent [27]. The paper emphasizes the importance of considering data structure in model validation and the selection of appropriate cross-validation strategies to avoid overly optimistic error estimates and potential overfitting.

In their work, Rabinowicz and Rosset delve into the intricacies of cross-validation (CV) for correlated data through a mathematical lens. They commence with the foundational equations of CV and progressively introduce assumptions to dissect various data scenarios [25]. Their analysis focuses on exploring biases, introduced by correlations in CV, leading to the development of a bias-corrected CV estimator, termed CVc. This estimator is particularly adept at providing an unbiased estimate of prediction error in numerous settings where standard CV falls short [25]. The robustness of their approach is exemplified by its application to two classical datasets: the Black Friday Dataset and the California House Prices Dataset. Their methodical approach offers a nuanced understanding of CV’s applicability and paves the way for more accurate model evaluation and selection in the presence of correlated data [25].

The next paper explores the challenges of evaluating financial time series forecasting models. The authors, Blake LeBaron and Andreas S. Weigend, critically examine the common practice of splitting data into fixed training, validation, and test sets [18]. They argue that this approach can lead to overly optimistic conclusions due to the lack of consideration for variability across different data splits. To address this, they propose a bootstrap method that combines the benefits of data splitting with resampling techniques, providing a more robust statistical analysis of forecast variability [18]. Their empirical study on the New York Stock Exchange trading volume data reveals two key findings: the variability due to resampling is significantly larger than that due to network conditions, and neural-network solutions with early stopping tend to closely resemble linear models, failing to extract significant nonlinearities [18]. The paper emphasizes the importance of not over-interpreting models based on a single data split and contributes to the understanding of model evaluation and uncertainty in financial forecasting.

The paper ”Does data splitting improve prediction?” by Julian J. Faraway, explores the efficacy of data splitting strategies in statistical model prediction [6]. The author investigates whether dividing data into separate sets for model selection and parameter estimation, referred to as data splitting, enhances the reliability of predictive distributions for future observed values. The paper contrasts the full data strategy, where all data is used for both model selection and estimation, with the data splitting approach, which reserves a portion of the data for model selection and the remainder for parameter estimation [6]. Faraway introduces a hybrid estimator, termed SAFE (Split Analysis, Full Estimate), which utilizes one part for model selection and the entire dataset for estimation. The study's simulations suggest that except in specific circumstances, data splitting strategies, particularly the SAFE approach, tend to outperform full data strategies for prediction purposes [6]. The paper provides a nuanced view of the trade-offs between model selection, parameter estimation, and data reuse costs, ultimately recommending a split data analysis when the model is not predetermined and a full data analysis in scenarios with a large number of parameters relative to observations [6].

Last but not least, Meng et al. present a critical examination of data-splitting strategies in the evaluation of recommender systems. The authors argue that the choice of data splitting strategy can significantly influence the ranking of recommender systems during evaluation [20]. They compare three common data splitting strategies—Leave One Last, Temporal Split, and Random Split—across seven state-of-the-art recommendation models on two datasets. Their findings reveal that data splitting is a confounding variable that can markedly alter system rankings, making much of the published literature non-comparable [20]. The paper emphasizes the need for standardization in evaluation methodologies to ensure fair and reproducible comparisons of recommender systems [20]. It also provides best practice recommendations for future research, including the reporting of splitting strategies and the use of standardized evaluation tools.

2.2. ML models

In this section, both classical ML models as well as advanced DL models are discussed in the context of NWP. The related work focuses on the evaluation techniques and not on the results. The last part presents some relevant literature studies on ML and DL models for NWP, along with their potential problems.

2.2.1. Classical ML models

Nalluri et al. present a study on weather forecasting using machine learning clustering techniques [23]. Their evaluation method is not clearly explained, as the authors only pointed that they divided the data into training and test sets without further explanations [23]. The research highlights the challenges in predicting weather due to its dynamic nature and the absence of a consistent target attribute.

The paper “Smart Weather Forecasting Using Machine Learning: A Case Study in Tennessee” presents a novel approach that leverages both spatial and temporal data for predictions [13]. Focusing on enhancing forecast accuracy for a target city, the study capitalizes on the radial influence of surrounding areas to infer weather conditions from the target city. The research employs historical data from the preceding two days to perform hourly forecasts [13]. The training set consists of 2 months of data and the test set is consisted of 7 days. This work consisted as an inspiration for this thesis, regarding the idea of using spatial splitting strategies as well.

In the paper “Machine Learning Applied to Weather Forecasting” by Holmstrom, Liu, and Vo, the authors explore the application of machine learning techniques to weather forecasting, a field traditionally dominated by physical models [10]. The study investigates the use of ML models to predict maximum and minimum temperatures over a seven-day horizon, based on two days of prior weather data [10]. The evaluation method they used was a classical one; they gathered weather data from 2011 to 2015 and they split the data as follows: the first four years as a training set and the last year as a test set. Moreover, ”a 4-fold forward chaining time-series cross-validation” [10] was executed.

In the paper “Bayesian Network Probability Model for Weather Prediction” Aye Nandar proposes a weather forecasting system for Myanmar using Bayesian networks (BNs) to model the spatial dependencies among meteorological variables [24]. While the approach is innovative in employing BNs to predict rainfall and temperature, the methodology is presented rather simply. Despite this, the author

used historical weather data from 1990 to 2005 to train and the year 2006 to test the model [24]. Moreover, the training data is split in 2 ways: one where the first 11 years are used as training and the last 5 years are used as validation and one where the 5 years of validation data are put in the middle of the training dataset.

In their work, Cofiño et al. present an innovative approach to probabilistic weather forecasting by employing BNs to model spatial and temporal dependencies among meteorological stations across the Iberian Peninsula [5]. Unlike traditional methods that treat stations independently, their model considers the interconnected nature of weather phenomena, where the conditions at other stations influence the rainfall prediction for one station [5]. The authors demonstrate the efficacy of their methodology through the successful prediction of precipitation across 100 stations during the winter of 1999 [5], and as training data, they used precipitation records from 1979 to 1993.

2.2.2. Advanced ML techniques

The next paper, ”Learning skillful medium-range global weather forecasting”, introduces a novel approach to predictive modelling, characterised by using an autoregressive system. This system is unique in its ability to feed its predictions back into the model as an input, creating a feedback loop that enhances the accuracy of future predictions [17]. The methodology employed in this research is based on graph neural networks, a type of neural network that excels in processing graph-structured data. The results of the study are noteworthy. The proposed method not only outperforms the current state-of-the-art High-Resolution Ensemble System (HRES), but it does so with a speed that sets a new standard in the field [17]. This work represents a significant contribution to the field of predictive modelling, particularly in terms of efficiency. In terms of data splitting, they gathered satellite data from 1979 until 2021, and they divided it as follows: the period from 1979 until 2015 was training data, 2016 and 2017 were used as validation data and the period from 2018 until 2021 was used as a test set.

The paper ”Predicting weather forecast uncertainty with machine learning” presents a Deep Learning approach utilizing Convolutional Neural Networks (CNNs), where the model’s architecture is designed to learn iteratively from historical errors and the spread of past weather predictions [28]. They gathered the Global Ensemble Forecasting System data from 1985 until 2016 and performed the following split: the last 7 years are used as test data, 2 other years as validation data and the rest as training data [28].

The research conducted by Goutham et al. reveals a clear geographical pattern in the accuracy of predictions, with coastal regions of France exhibiting different results compared to inland areas [8]. Crucially, the study identifies wind components such as wind speed, direction, and shear, along with atmospheric pressure, as the most significant variables influencing the models’ predictive capabilities [8]. For testing, they used a 10-fold cross-validation, where 9 folds are used for training and one fold is used for testing. Moreover, the performances are averaged over 10 repetitions [8].

Lai et al. proposed an early application of a deep learning approach to weather forecasting using a classical multi-layer perceptron (MLP) model [16]. Their work focused on short-term temperature and rainfall predictions over the east coast of China [16]. In terms of data splitting, they used meteorological data from 1999 until 2001, collected from several weather stations; although the authors did not give more details about how data was further split [16].

Li et al. developed a robust approach for probabilistic rain prediction using satellite data [19]. Their methodology leverages multi-band satellite images to predict rainfall, focusing exclusively on rainfall forecasting. As the authors participated in the 2022 NeurIPS competition, they tested their model multiple times, resulting in multiple data splits. In the first stage, they gathered satellite data from February until December 2019, and in the second stage, they had 2 years of training data (2019 and 2020) [19].

2.2.3. Review papers in NWP

Bochenek and Ustrnul provide an extensive literature review of 500 scientific articles published since 2018, focusing on machine learning applications in numerical weather prediction and climate analysis [3]. They highlight the prevalent research areas, such as photovoltaic and wind energy forecasting, atmospheric physics, and climate change studies, showcasing the integration of machine learning techniques in these fields [3]. The study offers valuable insights into various machine learning meth-

ods employed in the research, including Deep Learning, Random Forest, and Support Vector Machines, among others. The authors anticipate that machine learning will play a crucial role in advancing weather forecasting and climate research, suggesting potential future research trajectories in these domains [3].

In the paper “Can Deep Learning Beat Numerical Weather Prediction?” by Schultz et al., the authors embark on a comprehensive exploration of the potential for deep learning (DL) to revolutionize meteorological forecasting. They meticulously compare three distinct workflows: the traditional numerical weather prediction (NWP), a hybrid NWP-ML/DL approach, and an end-to-end DL methodology [29]. The study delves into various neural network architectures, assessing their applicability to the unique challenges of weather forecasting, such as autocorrelation, imbalanced data, missing values, and non-gaussian PDFs [29]. The authors also examine different strategies for splitting datasets into training, development, and testing sets, highlighting the significance of both probabilistic models and deep learning methods. This pioneering work serves as the cornerstone of the thesis, offering a critical evaluation of the current state and prospects of ML and DL in the realm of weather prediction.

The paper “Deep Learning and process understanding for data-driven Earth system science” underscores the pivotal role of deep learning techniques in enhancing predictive capabilities across various domains, notably meteorology. The authors highlight the unique challenges posed by weather data in the context of deep learning, such as interpretability, ensuring physical consistency, managing complex and uncertain datasets, scarcity of labelled data, and the computational intensity required [26]. The paper advocates for a synergistic integration of deep learning with physical models, suggesting avenues for improvement including parameterization refinement, substituting machine learning models for physical sub-models, analyzing discrepancies between models and observations, constraining submodels, and employing surrogate modelling [26]. Concluding with forward-looking research directives, the paper calls for recognition of data particularities, ensuring plausible and interpretable inferences, robust uncertainty estimation, and rigorous validation against intricate physical models [26].

The study by Moosavi et al. presents a significant advancement in quantifying uncertainty through machine learning techniques. The paper delves into the mathematical underpinnings of model error estimation and identifies two primary challenges: the prediction of model errors in advance for specific aspects of interest, and the determination of physical packages that significantly influence forecast uncertainty [22]. Through rigorous experimentation, the authors compare the efficacy of Random Forests and Artificial Neural Networks (ANNs) in addressing these challenges. Their findings suggest that ANNs exhibit greater accuracy in forecasting, thereby offering a promising avenue for enhancing the precision of NWP models [22]. This study not only contributes to the understanding of uncertainty quantification in NWP but also underscores the potential of machine learning algorithms in improving the reliability of weather forecasts.

2.3. In situ vs. satellite data

This section underlines the comparison between *in situ* and satellite data, both on results and evaluation methods. Some of the papers do not split the data in general, as the comparison is done with all the gathered data.

Cho et al. focused on exploring various predictive models, including random forest, support vector regression, artificial neural network, and a multi-model ensemble, to analyze both spatial data from Local Data Assimilation and Prediction System (LDAPS) and *in situ* observations [4]. While *in situ* data provides detailed insights at specific locations, LDAPS data compensates for the lack of spatial continuity in ground-based observations, offering a broader perspective on environmental conditions. Concerning the evaluation methods, the authors took the period from 2015 to 2017 for training and testing. Practically, they took a day for training and the next day for testing.

The next paper, ”Review on Spatio-Temporal Solar Forecasting Methods Driven by In Situ Measurements or Their Combination with Satellite and Numerical Weather Prediction (NWP) Estimates”, provides a comprehensive analysis of spatio-temporal solar forecasting methods, particularly focusing on the use of *in situ* measurements and their combination with satellite and NWP estimates [1]. While the paper’s scope extends beyond solar forecasting for photovoltaic energy, it dives into a large variety of methods, including statistical methods, classical machine learning techniques and advanced deep learning approaches. The paper emphasizes the distinction between *in situ* data, which offers

higher resolution and bypasses the uncertainty of converting cloud images to irradiance forecasts, and satellite data, which provides broader spatial coverage but with associated latency in data acquisition and processing [1]. This differentiation is crucial for understanding the operational characteristics and optimal forecast horizon range of each data type in solar forecasting applications. As it is a review, there are no technical evaluation methods (e.g. splitting the dataset into train and test). The review's findings highlight the potential of integrating diverse data sources and methods to enhance forecasting accuracy and suggest directions for future research in this rapidly evolving field.

The study conducted by Wang and Prigent stands out for its meticulous comparison of Land Surface Temperature (LST) estimates derived from four distinct datasets—two satellite-based, one *in situ*, and one reanalysis from Numerical Weather Prediction (NWP)—across diverse climates and regions [33]. A critical aspect of the study was the evaluation of satellite data's concordance with *in situ* measurements. The findings revealed that discrepancies were more pronounced during warmer seasons and around midday [33]. The comparison highlighted that the congruence of the European Centre for Medium-Range Weather Forecasts (ECMWF) reanalyses with LST observations varied with land cover; notably, the reanalysis dataset aligned more closely with *in situ* data over barren land, especially at night, attributed to advancements in the surface model [33]. This study underscores the importance of considering seasonal, diurnal, and land cover variations when assessing the accuracy of satellite-derived LST estimates against *in situ* measurements.

In their 1986 study, Isaacs et al. explored the integration of *in situ* data and satellite observations for NWP models. They utilized a range of parameters including temperature, cloudiness, precipitation, wind, and surface properties, alongside satellite data spanning wavelengths from 0.6 µm to 1.35 cm [12]. The paper delves into both the inverse problem—deducing the most fitting model from a set of observations—and the forward problem—forecasting observations based on a given model. Despite the comprehensive approach, the results were not as definitive as anticipated [12]. The study noted the potential for retrieving other meteorological parameters beneficial for NWP, such as surface winds from microwave sensors and cloud drift winds aloft, in addition to surface properties like soil moisture, albedo, snow cover, temperature, and fluxes [12]. This early research highlights the contrast between *in situ* measurements, which provide localized, direct observations, and satellite data, offering broader, indirect measurements, underscoring the challenges in effectively combining these data sources for enhanced weather prediction.

The study presented in the paper "Comparison of satellite-derived and in-situ observations of ice and snow surface temperatures over Greenland" provides a comprehensive analysis of land surface temperatures (LSTs) obtained from three satellite instruments and compares them with *in situ* observations from the Greenland Climate Network [9]. The research highlights the reliability and consistency of satellite-derived LSTs under clear-sky conditions and relatively flat terrain, across a temperature range of −40 to 0 °C. Notably, the study found that while there were instances of significant discrepancies between satellite and *in situ* data, with differences up to 3°C, other comparisons showed remarkable agreement [9]. These findings underscore the potential of satellite instruments to provide accurate LST measurements, although the study also emphasizes the importance of considering the inherent variability and limitations when comparing these measurements with point-based *in situ* data.

2.4. Clustering

In their influential paper, Blondel et al. present a groundbreaking algorithm for the fast unfolding of community structures in large networks [2]. The algorithm (later named as Louvain algorithm), which significantly advances the state of the art in network science, is based on the optimization of modularity—a measure that quantifies the quality of a division of a network into communities [2]. It operates through a two-phase, iterative process. In the first phase, each node is assigned to its community. The algorithm then considers each node in the network and evaluates the gain in modularity by moving it to the community of each of its neighbours, placing the node in the community that results in the highest modularity gain. This process is repeated until no further improvements can be achieved [2].

The second phase involves creating a new network where nodes represent the communities identified in the first phase. The weights of the links between these new nodes are determined by the sum of the weights of the links between nodes in the original communities. This effectively reduces the size

of the network, and the two-phase process is repeated iteratively on this smaller network [2]. This hierarchical approach continues until no further modularity improvements are possible, producing a dendrogram that reveals the hierarchical structure of the network.

Blondel et al.'s algorithm is notable for its computational efficiency, which is achieved through its localized update strategy and the reduction of the network size in successive iterations. The authors validated their method on several large-scale networks, including a Belgian mobile phone network with 2 million customers and a web graph containing 118 million nodes and over a billion links, showcasing the algorithm's ability to handle extremely large datasets [2]. Furthermore, the algorithm has been shown to outperform other community detection methods in terms of both speed and the quality of the communities detected. The method's scalability and effectiveness make it a valuable tool for analyzing the structure of large and complex networks, providing insights into their modular organization that are crucial for understanding various real-world systems.

3

Experimental setup

In this section, we will discuss first the general framework, and then explain the most complex parts for both temporal and spatial splitting strategies.

3.1. General framework

The general framework for all the temporal strategies, except T_a is shown in the figure 3.1. It is worth mentioning that the first step ("Select a single feature of the dataset") is catered to our experiment, although more features can be used in this framework. The key difference between strategy T_a and the other strategies lies in the order of the steps. In strategy T_a, sliding windows are created before splitting and normalization, meaning that initially, the windowing process works with the raw feature data. In contrast, the other strategies split and normalize the data first, and only then the sliding windows are created, ensuring that each window contains pre-processed data. This difference is significant, because when windowing there is a risk of data leakage, and splitting must first be done. Moreover, for strategy T_a, the random rows need to be sampled on the windowed data, to ensure coherence.

Figure 3.1: General framework for evaluating temporal data splitting strategies.

The general framework for all the spatial strategies is shown in figure 3.2. The main difference between the temporal and the spatial strategies is that for the latter, one needs to make the clusters before employing all the strategies. Before clustering, distances between cities need to be computed. To ensure coherence, the sliding windows are created first, and then the data is split in accordance with the strategies. Therefore, data normalization is applied at the end of the pipeline.

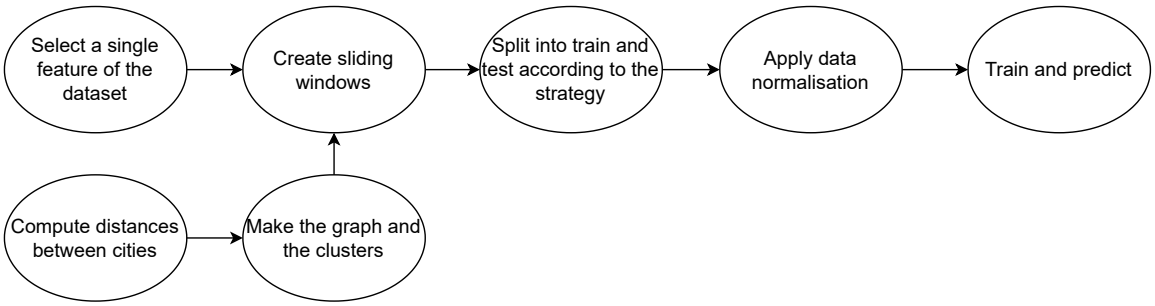


Figure 3.2: General framework for evaluating spatial data splitting strategies.

3.2. Time-series model

Suppose a time-series $X = \{x_t\}_{t=0}^{T}$, where x_t is current value, t is the current time and T is the last time. Considering the dataset as a time-series, conversion to a supervised setting was needed and therefore, a sliding window was implemented. Mathematically, the sliding window looks as such:

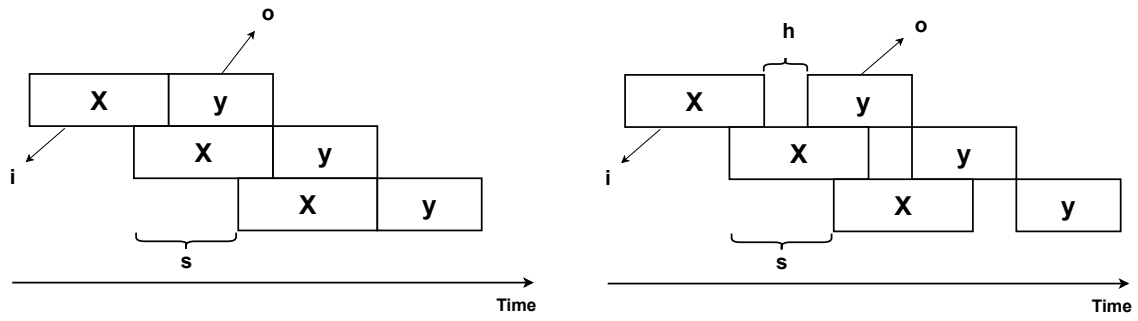
$$X = \{x_{t:t+i}\}_{t=0}^{T-i-o};\ y = \{x_{t+i:t+i+o}\}_{t=0}^{T-o},\ t = \{0, s, 2s, ...\},\ i, o, s = \{1, 2, 3, ...\}$$

where i represents the input window size, o represents the output size and s represents the step size; all three can have different sizes, although, for the experiment, they were all set to 7. Furthermore, X represents the feature matrix and y represents the target matrix. One week is fed into the model, with the task being to predict the following week. Thus, the feature matrix contains 7 columns, and the target matrix has 7 columns as well. The graphical representation of the sliding window can be seen in figure 3.3a.

Another case is when a week is fed into the data, and then another week in the future (not necessarily the next one) is predicted. The difference in time between the two is called a horizon, and it is an important factor in determining the difference between these data-splitting strategies. Mathematically, the sliding window with the horizon looks as follows:

$$X = \{x_{t:t+i}\}_{t=0}^{T-i-h-o};\ y = \{x_{t+i+h:t+i+o+h}\}_{t=0}^{T-o},\ t = \{0, s, 2s, ...\},\ i, h, o, s = \{1, 2, 3, ...\}$$

where i represents the window size, o represents the output size, s represents the step size and h represents the horizon size. The graphical representation of the sliding window with horizon can be seen in figure 3.3b.



(a) Sliding window without a horizon

(b) Sliding window with a horizon

Figure 3.3: Graphical representation of sliding windows

This time-series model and the sliding window were implemented in temporal and spatial strategies.

3.3. Data splitting strategies

This experiment assesses four temporal data splitting strategies and three spatial data splitting strategies. Before listing the temporal splitting strategies, it is worth mentioning that all the mathematical notations are only for the testing data (and possibly, validation data), as training data can be derived by subtracting the testing data from the whole dataset. The temporal data splitting strategies are:

- T_a: A random splitting of data into train and test; namely random samples from the dataset are taken as a test set, while the rest of the data is used for training the model. The mathematical formulation for this strategy is: $T_a = \{x_t\}_{t=random}$, where x_t is the value of the data at the data point t. These points can be single time points or they can form small windows, depending on how many points are taken for the testing data.
- T_b: The last year of data is used for testing and the rest of the dataset is used for training. The mathematical notation for this strategy is $T_b = \{x_t\}_{t=T-365}^{T}$, where x_t is the value of the data at the data point t, and T is the last data point. The standard year is formed by 365 data points, assuming that every day, exactly one measurement is taken.

- T_c: One of the years is randomly selected from the dataset for testing, and the remainder of the dataset is taken as training. The mathematical notation for this strategy is $T_c = \{x_t\}_{t=(n-1)\cdot 365}^{n\cdot 365}$, where x_t is the value of the data at the data point t and n is the year number in the dataset, starting from 0.
- T_d: The test data comprises separate time blocks that add up to one year. These separate blocks are randomly placed in the dataset, and the remainder of the dataset is used for training. The mathematical model for this strategy is: $T_d = \{x_t\}_{t=z}$, where x_t is the value of the data at the data point t, and z represents the different seasons. A season can be defined as a continuous block of time, smaller than a year (e.g. 3 months) or the period previously defined.

The spatial splitting strategies are:

- S_a: A random splitting of the whole dataset (all weather stations) into train and test is performed. Random samples from the dataset are taken as a test set, while the rest of the data is used for training the model.
- S_b: Clusters of weather stations are formed and one of them is taken as a test set and the others as a training set. These clusters are formed by their geographical localisation, and therefore, the number of clusters can differ from one graph to another. The number of clusters did not matter when performing the experiments, as the distances and the connectivity of a node were more important.
- S_c: Clusters of weather stations are formed. One of them is taken as a test set and all non-neighbouring clusters are taken as training sets.

3.4. Preprocessing for spatial data

For the spatial splitting strategies, it is assumed that the coordinates of the stations are priorly known, to compute the distances between all the weather stations. The distance metric used was the Haversine distance, which takes two coordinates (both latitude and longitude) and taking into account the spherical shape of Earth, then computes the distance between the two coordinates. With an error of a maximum of 0.5% [7], it is a fairly accurate formula for computing distances on Earth. The equation 3.1 shows the actual formula, where φ_1, φ_2 are the latitudes for the first and, respectively, the second coordinate and λ_1, λ_2 are the longitudes for the first and, respectively, the second coordinate, d represents the distance and r represents the radius of the sphere.

$$d = 2r \cdot arcsin\left(\frac{\sqrt{1 - cos(\varphi_2 - \varphi_1) + cos\varphi_1 \cdot cos\varphi_2 \cdot (1 - cos(\lambda_2 - \lambda_1))}}{2}\right) \tag{3.1}$$

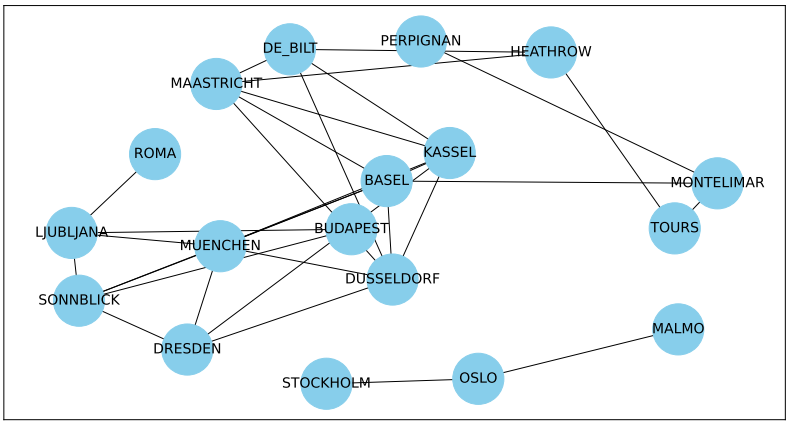


Figure 3.4: Spatial distribution of the dataset.

For creating the graph with the stations, a distance threshold is fixed as low as 30 km and as high as 1000 km. That means if the stations are further than the threshold, in the graph they appear as having no direct connection. In this experiment, a threshold of 500 km was used, considering the data limitations: the weather stations were spread across half of Europe and there were only 18 stations taken into consideration. The selection of the threshold was empirical, seeing that the resulting graph would be sparse if a lower threshold was chosen. If a higher threshold had been chosen, then the graph would be almost fully connected and the clustering would not have been possible anymore. Ultimately, a clustering was made on this graph by running the Louvain algorithm [2], presented in Chapter 2. Louvain algorithm was selected for the clustering task for multiple reasons. First of all, the generality played an important role, as the algorithm can be used in other datasets without changing the number of clusters. Secondly, for the task, the number of clusters is not needed, as they form regions based on the distance and the connectivity of the graph.

3.5. Datasets

One dataset was selected from Kaggle to perform the experimental test for both temporal and spatial splitting strategies. The dataset is called "Weather Prediction" and contains weather data from 18 weather stations in Europe throughout the years 2000 to 2009, the last data point being 1st January 2010 [11, 14]. It contains also some meteorological features such as maximum and minimum temperature, humidity, precipitation, solar radiation, etc.

To verify whether the data or the model behaves properly, an artificial dataset for the maximum temperature was created. Only the maximum temperature feature was created because other meteorological features should have been defined by more complex mathematical functions (e.g. impulse functions). From the definition of the dataset, $X = \{x_t\}_{t=0}^{T}$ depending on the size of the data, $T = 2 \cdot n \cdot \pi$, where n is the number of years taken into account; in our experiment, $n = 10$ and $n = 2$. We chose these amounts of years for our experiments for two reasons: to resemble the number of years in the Kaggle dataset and to have an extreme situation where the amount of data is scarce.

$$x_t \; = \; Norm(12 \cdot sin(t-2) \; + \; 15 \; + \; c \cdot t) + \varepsilon, \; \varepsilon \sim \mathcal{N}(0, \; \sigma) \tag{3.2}$$

The artificial dataset is modelled as a sine function: $x_t = 12 \cdot sin(t-2) + 15$. This artificial dataset is only catered for the the maximum temperature, as the other features are not described as sine functions, but rather impulse functions. In addition to this function, two parameters were added to find a potential difference between the temporal strategies. One of them is the Gaussian noise, which is added after data normalization, as follows: $\varepsilon \sim \mathcal{N}(0, \; \sigma)$, where ε is the noise, and σ is the standard deviation. Moreover, next to the seasonal change, a linear trend is added to the model for climate change. This renders the following equation 3.2.

Eventually, the artificial data was created following the period of the real data. Therefore, 3654 data points were created, for $n = 10$ case and 732 data points for $n = 2$ using the equation 3.2. Each data point represents a day, and as an example, years from 2000 to 2009, the same as in the real data, were taken into consideration, as illustrated in figure 3.5.

As previously mentioned, n takes two values: 2 and 10; this means that 2 two regimes are introduced: one with low data and one with high data. In both regimes, the strategies were evaluated under various scenarios to understand their performance and robustness. In the first scenario, the artificial data was tested without considering horizon or climate change influences (i.e. $c = 0$). This baseline provided a reference for understanding how the strategies perform under stable conditions. In the second scenario, the artificial data incorporated elements of climate change, allowing for the assessment of how well the strategies adapt to changing patterns in the data. The third scenario introduced only the horizon effect, focusing on how the strategies handle time-dependent variations. The fourth scenario combined both horizon effects and climate change, presenting the most complex and realistic challenge for the strategies.

On both real and artificial datasets a data normalization was conducted. Normalization is fitted on training data, resulting in a mean of 0 and a standard deviation of 1. The learned parameters of the

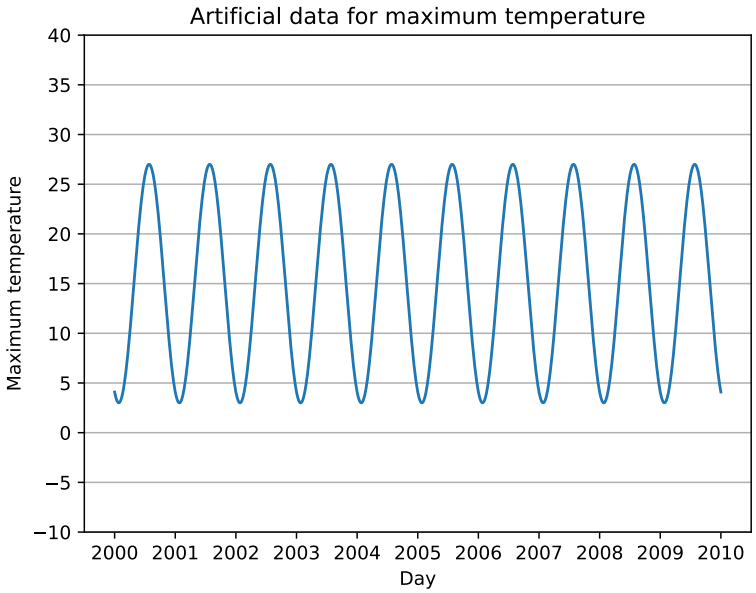


Figure 3.5: Artificial data

normalization were then applied to test data. As previously mentioned, after data normalization, noise was added to the time series.

The data processing and the time-series model are the same for the real dataset, only that it was not generated by equation 3.2. For spatial data splitting strategies, only real data was used, and the sliding windows implementation is the same.

4

Results

This chapter showcases the results obtained by the splitting strategies. Linear regression was used as an ML model and the numerical results are computed with RMSE. The RMSE was run multiple times to have a mean and a standard deviation. The difference between the train and test sets is also presented.

4.1. Temporal splitting

This section underlines the results of the temporal splitting strategies. First, the results for the real data are shown and then for the synthetic data on both high and low regimes.

4.1.1. Results for Europe dataset (real data)

For the experiments, 4 meteorological features were considered (maximum temperature, humidity, precipitation and atmospheric pressure). These features were not put together in this experiment, because the experiment was designed to be simple. Furthermore, 5 weather stations were assessed, depending on their geographical location within Europe (one in every cardinal point, plus one in the centre). The decision was made to underline the eventual differences between the distinct regions and climates within Europe. Moreover, the horizon is incrementally increased to see a tentative difference between the temporal strategies. Theoretically, an increase in the horizon means an increase in the gap between the given data and the prediction that needs to be made. From now on, the Europe weather dataset will be referred to as the real data.

	Basel weather station				**Oslo weather station**			
	T_A	T_B	T_C	T_D	T_A	T_B	T_C	T_D
Train	0.51 (0.01)	0.51 (0.02)	0.52 (0.01)	0.52 (0.01)	0.42 (0.01)	0.41 (0.01)	0.42 (0.01)	0.42 (0.01)
Test	0.53 (0.02)	0.55 (0.04)	0.52 (0.03)	0.52 (0.03)	0.44 (0.02)	0.46 (0.07)	0.43 (0.05)	0.42 (0.04)
Δ	0.02 (0.02)	0.03 (0.05)	0 (0.03)	0 (0.03)	0.02 (0.02)	0.05 (0.07)	0 (0.05)	0 (0.04)

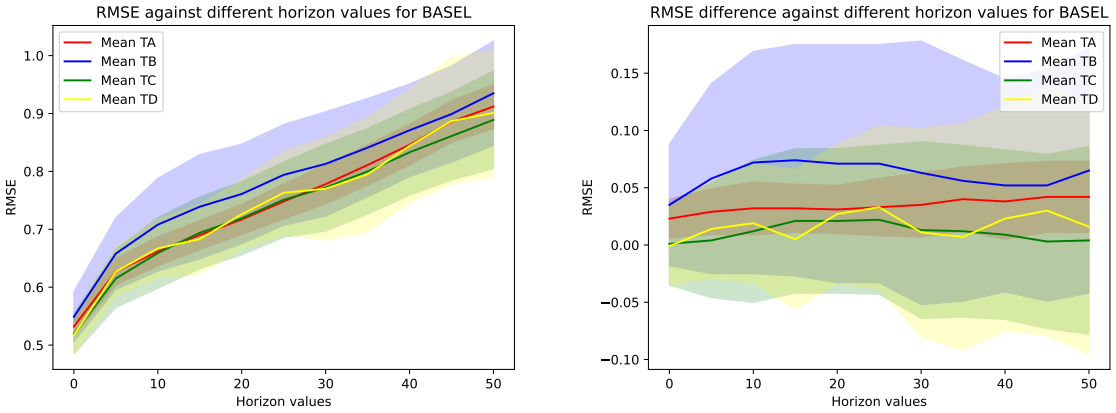
Table 4.1: RMSE results on weather prediction for different scenarios for Basel and Oslo weather stations. In this case $h = 0$ and Δ represents the difference between test and train RMSE

	Basel weather station				**Oslo weather station**			
	T_A	T_B	T_C	T_D	T_A	T_B	T_C	T_D
Train	0.88 (0.01)	0.87 (0.03)	0.89 (0.01)	0.89 (0.01)	0.86 (0.01)	0.86 (0.05)	0.87 (0.01)	0.86 (0.01)
Test	0.91 (0.04)	0.94 (0.09)	0.89 (0.08)	0.9 (0.12)	0.9 (0.04)	0.93 (0.13)	0.86 (0.08)	0.82 (0.12)
Δ	0.04 (0.03)	0.07 (0.11)	0 (0.08)	0.02 (0.12)	0.04 (0.03)	0.07 (0.17)	0 (0.08)	-0.05 (0.1)

Table 4.2: RMSE results on weather prediction for different scenarios for Basel and Oslo weather stations. In this case $h = 50$ and Δ represents the difference between test and train RMSE

The samples were associated with strategies for computing the RMSE mean and standard deviation while considering the data size limitations, as the dataset contains 10 years of data. Because the

dataset is not large enough (only 10 years) to ensure both equality between the number of samples used and the stability of the splitting strategy, we decided to use 100 samples for strategy T_A and T_D, only 9 samples for strategy T_B and only 10 samples for strategy T_C. The results for 2 weather stations: Basel and Oslo are shown in the tables 4.1 and 4.2 and the figures 4.1a, 4.1b, 4.2a, 4.2b as a demonstration and comparison of the different Europe climates as well. The results for the other stations are found in the appendix A.



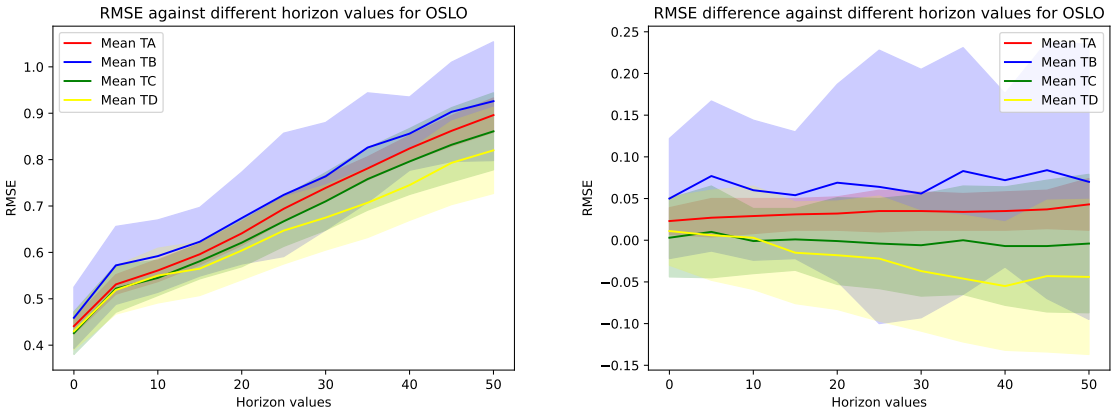
(a) Test RMSE

(b) Difference between train and test RMSE

Figure 4.1: RMSE results against different h values for Basel station on maximum temperature feature

Relatively to the weather station, the performance of the data splittings changed slightly. Among the temporal strategies, strategy T_A has the lowest RMSE standard deviations and, therefore, is the most stable strategy. This also applies to trends, where T_A can follow the general RMSE trends on the horizon, without many fluctuations. Opposite to that, strategy T_B exhibits the most uncertainty, having a large standard deviation.

An interesting note is that when looking at the mean difference between the train and the test sets in the Oslo station, one can see that the metric becomes negative for T_D strategy when the horizon increases, which suggests that in this case, there is no overfitting. In another train of thought, the standard deviation of the difference for T_A and T_D is much larger than the others, despite possibly having more samples.



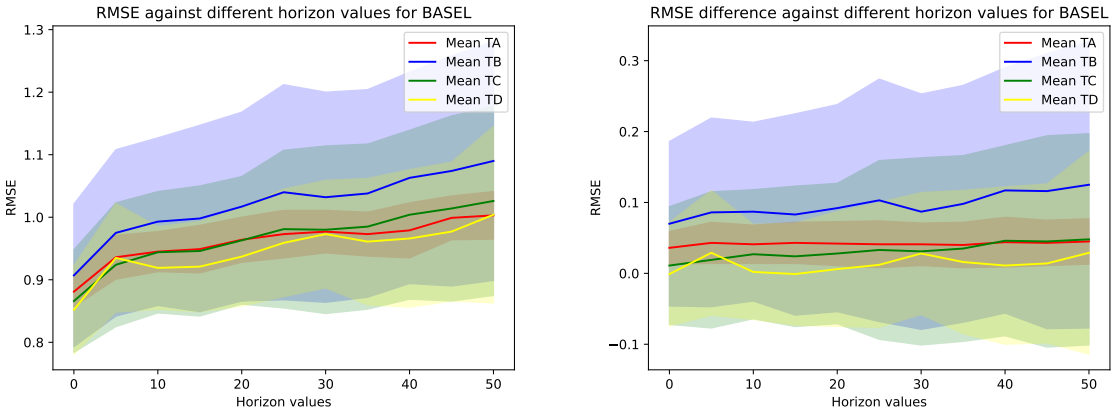
(a) Test RMSE

(b) Difference between train and test RMSE

Figure 4.2: RMSE results against different h values for Oslo station on maximum temperature feature

In the analysis of real-world data, our findings indicate that Strategy T_A consistently produces more stable results, suggesting a higher level of robustness to variations in the data. There are two reasons

for this phenomenon: the number of samples taken for T_A was high enough to ensure stability and the fact that random points are taken can secure stability in the results. However, it is important to note that Strategy T_A tends to overfit, as evidenced by the discrepancies between training and testing performance. This is the other side of having the random strategy in a time series, where data is auto-correlated. On the other hand, Strategy T_C demonstrates resilience against overfitting across both small and large forecasting horizons, though it does not achieve the same level of stability as Strategy T_A. The main cause of the lower stability of T_C is that there were not as many samples as in the Strategy T_A.

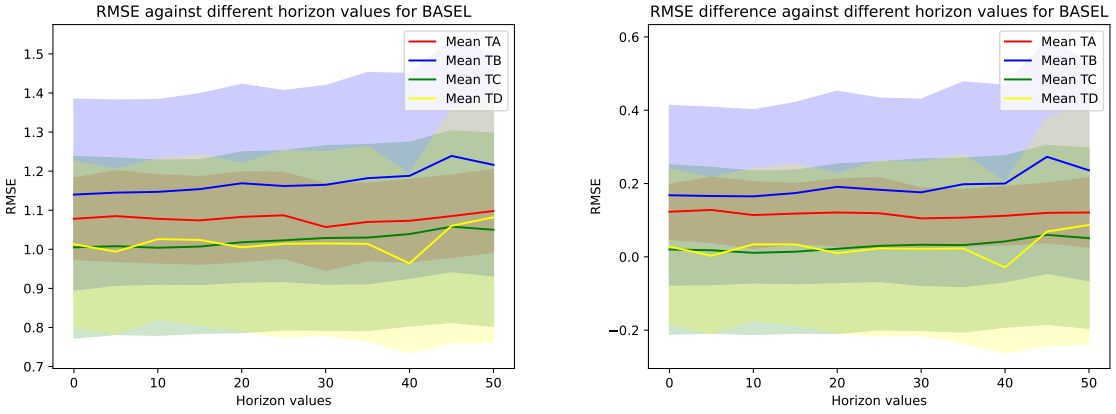


(a) Test RMSE

(b) Difference between train and test RMSE

Figure 4.3: RMSE results against different h values for Basel station on humidity feature

In contrast, Strategy T_B is generally not recommended due to its proneness to overfitting and its lower stability. Although being the most frequently used strategy, this combination makes it a less reliable option in practical applications. One cause of the poor results is that the number of samples assigned to strategy T_B is less than the samples assigned to other strategies. Another cause of the poor results is the length of the training dataset being changed from sample to sample, ranging from one year to 9 years. Furthermore, the performance of Strategy T_D appears to be highly contingent upon the specific climatic conditions of the station being analyzed. For instance, the effectiveness of Strategy T_D differs significantly between stations located in Basel (Central Europe) and Oslo (Northern Europe) because the temperatures in Northern Europe can follow a less noisy pattern than the temperatures in Central Europe.



(a) Test RMSE

(b) Difference between train and test RMSE

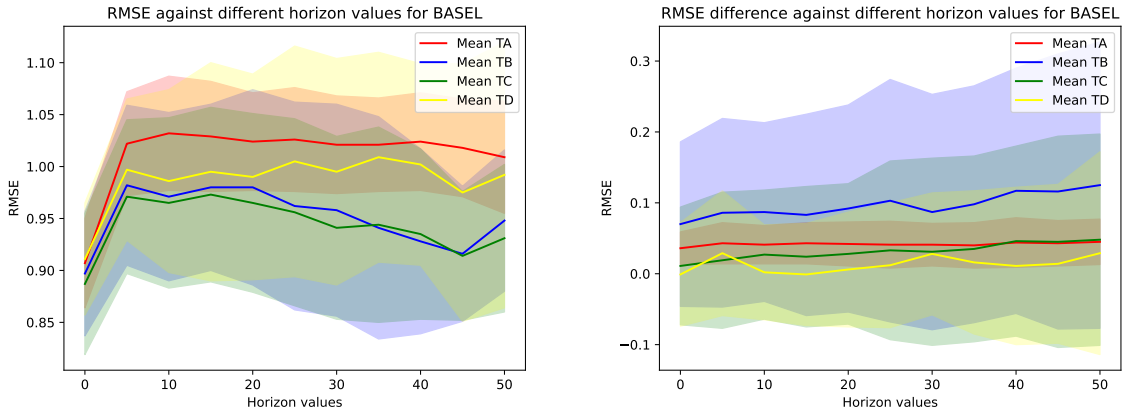
Figure 4.4: RMSE results against different h values for Basel station on precipitation feature

For this feature, all the other results, figures and tables are in the appendix A. The reader should expect

the same general results.

Regarding non-seasonal data, figures 4.3a, 4.3b, 4.4a, 4.4b show that the general trend follows the temperature feature with some differences. The error values are higher, generally around or above 1, whereas for the temperature feature, the RMSE was always below 1. Another aspect worth mentioning is that the trend for test data results is more steady.

Strategy T_A has the least standard deviation of the mean RMSE for precipitation and humidity, making it one of the most stable data splitting strategies. This is not the case for strategy T_B, which has a large standard deviation of the mean RMSE for all features. On top of that, strategy T_B has a higher error and a higher difference between train and test on humidity and precipitation. Although not significant, the mean errors of the strategies T_C and T_D are consistently lower than the other strategies.



(a) Test RMSE

(b) Difference between train and test RMSE

Figure 4.5: RMSE results against different h values for Basel station on atmospheric pressure feature

Atmospheric pressure results show strategy T_A has the highest error, although the difference is not significant. Strategies T_B and T_C have virtually the least RMSE error on test data, making them more recommendable in this situation. However, the difference between train and test is slightly higher in strategy T_B, making it less desirable to use.

The results for the other cities are in the appendix A.

4.1.2. Results for synthetic dataset (artificial data)

This section shows the results for both high data and low data regimes, with all the scenarios discussed in section 3.

High data regime

For a fair check, 10 years of data were taken to have the same amount of data as the Kaggle dataset. Furthermore, all four scenarios were assessed, although the results for the second scenario ($c = 0.2, h = 0$) were similar to the results of the first scenario, and therefore not shown here. The table 4.3 and the graphs 4.6a, 4.6b, 4.7a, 4.7b, 4.8a, and 4.8b show the RMSEs for train and test data and the differences between them (the default values were $h = 7$, $c = 0.2$, and $\sigma = 0.2$). For the high data regime, the number of samples is the same as for the real data.

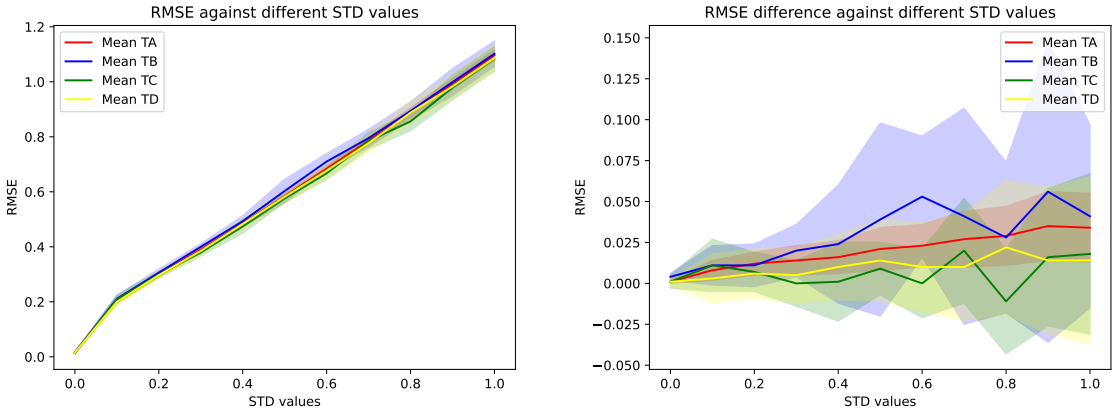
The findings from the artificial data largely support those obtained from real-world data, reinforcing the reliability of the dataset for testing purposes. The consistency between artificial and real data results suggests that the dataset is free from significant biases or inconsistencies, thereby validating its use for subsequent analyses. This agreement is crucial for ensuring that the conclusions drawn from the tests are dependable and reflective of true performance characteristics.

The temporal strategies are quite the same in the majority of scenarios, although some notes are worth to be taken. As a general trend, when the horizon increases, there is a higher chance that the strategies will underfit: when $h = 0$, the RMSE is comparable with the standard deviation of the noise and the

	$c=0, h=0$				$c=0, h=50$				$c=0.2, h=50$			
	T_A	T_B	T_C	T_D	T_A	T_B	T_C	T_D	T_A	T_B	T_C	T_D
Train	0.54 (0.01)	0.53 (0.02)	0.54 (0.01)	0.54 (0.01)	0.95 (0.02)	0.94 (0.02)	0.95 (0.01)	0.95 (0.01)	0.91 (0.02)	0.9 (0.04)	0.89 (0.01)	0.9 (0.01)
Test	0.56 (0.01)	0.57 (0.02)	0.55 (0.02)	0.55 (0.02)	0.98 (0.03)	0.93 (0.03)	0.96 (0.04)	0.96 (0.06)	0.94 (0.03)	0.99 (0.04)	0.91 (0.05)	0.92 (0.09)
Δ	0.02 (0.01)	0.03 (0.03)	0 (0.02)	0.01 (0.03)	0.04 (0.03)	-0.01 (0.04)	0.01 (0.05)	0.01 (0.06)	0.04 (0.03)	0.09 (0.07)	0.02 (0.05)	0.02 (0.1)

Table 4.3: RMSE results on different scenarios for high data regime. The σ is 0.5 in all settings; c represents the climate change coefficient, h represents the horizon and Δ represents the difference between test and train RMSE

differences between train and test are not that high. When increasing it, the strategies begin to underfit: both training and test errors are much higher than the standard deviation of the noise and an increase in the differences is seen as well.

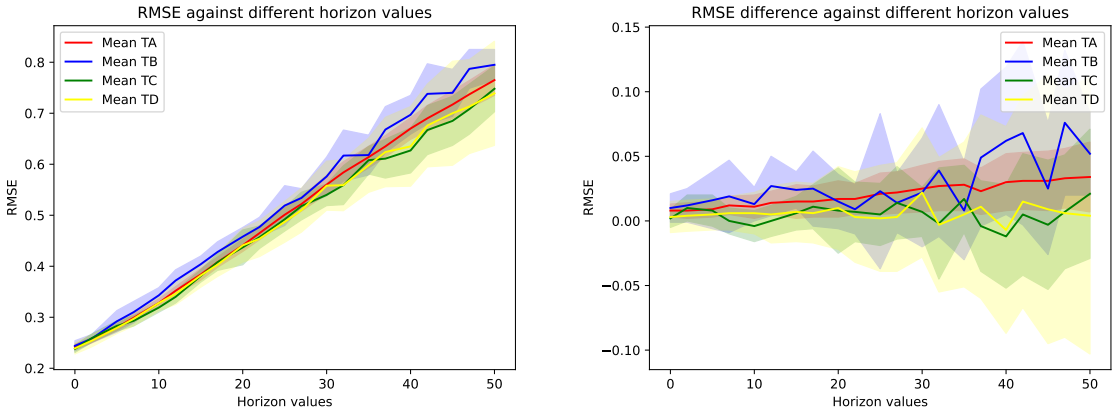


(a) Test RMSE

(b) Difference between train and test RMSE

Figure 4.6: RMSE results on the high data regime against different σ values, $h = 7,\ c = 0.2$

It is important to notice that strategy T_B is more unstable and has higher RMSE than its counterparts when c gets larger, as seen in the figure 4.8a. This issue is caused by the smaller number of samples for the strategy T_B. In this scenario, all the other three temporal strategies have insignificant differences. Nevertheless, when discussing the difference between training and test, figure 4.8b shows strategy T_B is more unstable in all scenarios, although the mean differences are not much higher than 0.



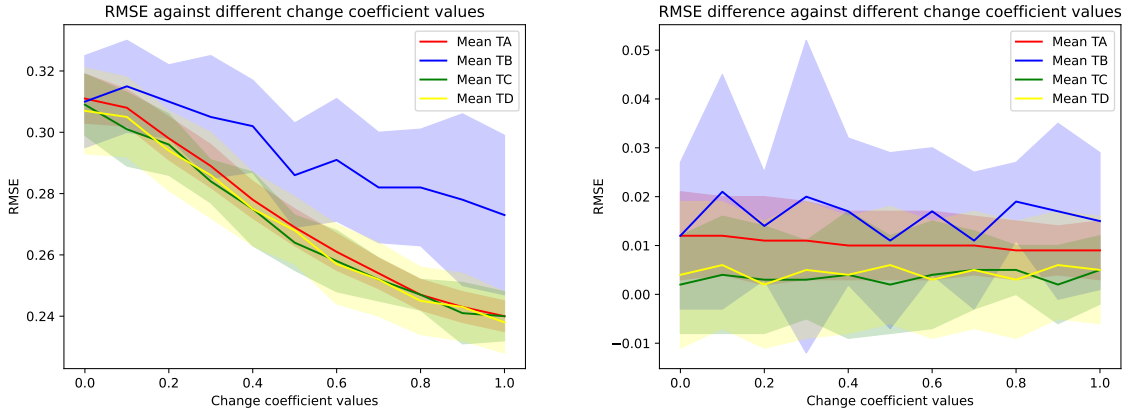
(a) Test RMSE

(b) Difference between train and test RMSE

Figure 4.7: RMSE results on the high data regime against different h values, $\sigma = 0.2,\ c = 0.2$

When $c = 0;\ h = 50$, in the strategy T_B, the RMSE difference between test and train is negative, which suggests that the strategy is better performing on test data. Even so, an unusual underfitting is exhibited, because the RMSE is about 0.9, but the added noise has a standard deviation of just 0.5. This phenomenon is caused by the high horizon as practically, the model needs to predict the maximum temperature 50 days ahead.

One interesting trend in this regime is that all strategies have lower test RMSE when it comes to increasing c, while the difference between training and test is the same, whether low or high c. In this regime, the difference in RMSE between $c = 0$ and $c = 1$ is almost 0.1, which suggests that the model begins to learn the overall trend in increasing the maximum temperatures. On the other side, the RMSE for strategy T_B is not decreasing at the same pace as the other strategies due to some samples from this strategy having less training data than their counterparts. While strategies T_C and T_D have a lower mean difference between the train and test sets, these differences are more unstable than their counterparts.



(a) Test RMSE

(b) Difference between train and test RMSE

Figure 4.8: RMSE results on the high data regime against different c values, $\sigma = 0.2,\ h = 7$

In the end, in the context of the high data regime, where an ample amount of data is available, strategy T_C emerges as particularly effective. This strategy exhibits the least difference between training and testing performance, indicating a minimal tendency towards overfitting. Overfitting, which occurs when a model performs well on training data but poorly on unseen test data, is a common concern in machine learning. Therefore, strategy T_C is recommended when overfitting is a major consideration in selecting a data-splitting approach. Additionally, if performance stability is prioritized, strategy T_A stands out as the most stable across both short and long-forecasting horizons. This stability ensures that the performance of the model remains consistent regardless of the prediction timeframe, which is critical for reliable forecasting.

Low data regime

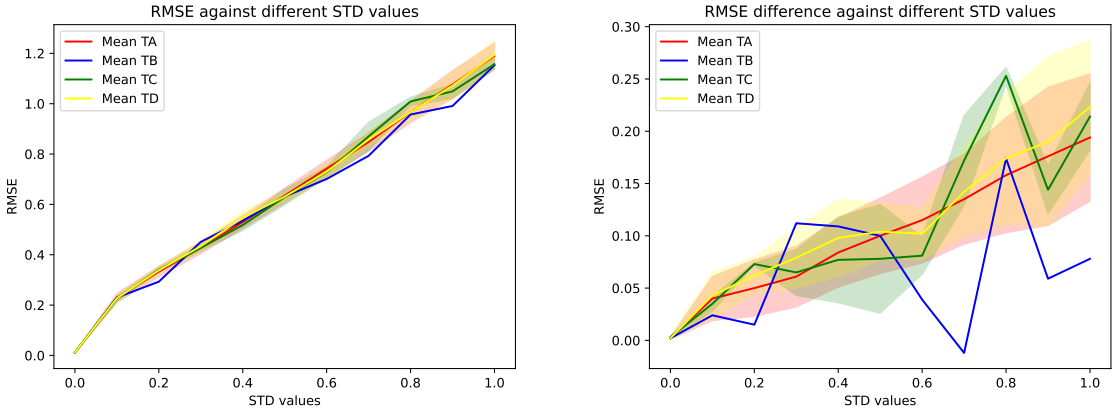
To investigate the impact of data availability and to exaggerate the differences between the strategies, the dataset was reduced to only two years of artificial data. This consisted of one year of training data and one year of testing data, with no validation data included.

Through these varied conditions, the differences between the strategies became more pronounced, highlighting how each approach copes with limited data and different environmental factors. This thorough evaluation helps in understanding the strengths and weaknesses of each strategy in a controlled yet challenging setting.

The results for each of the settings can be seen in the table 4.4. The eventual differences between the temporal strategies can be seen in the figures 4.9a, 4.9b, 4.11a, 4.11b, 4.10a, 4.10b (the default values were $h = 7$, $c = 0.2$, and $\sigma = 0.2$). Because the amount of data is different than in the high data regime, the number of samples taken for each strategy differs. Therefore, 100 samples were taken for strategy T_A, 1 sample for T_B, 2 samples for T_C and 16 samples for T_D.

	$c = 0, h = 0$				$c = 0, h = 50$				$c = 0.2, h = 50$			
	T_A	T_B	T_C	T_D	T_A	T_B	T_C	T_D	T_A	T_B	T_C	T_D
Train	0.5 (0.02)	0.47	0.51 (0.03)	0.5 (0.02)	0.9 (0.04)	0.8	0.86 (0.05)	0.82 (0.08)	0.9 (0.04)	0.78	0.82 (0.08)	0.78 (0.05)
Test	0.59 (0.03)	0.6	0.6 (0.01)	0.6 (0.03)	1.08 (0.08)	1.04	0.98 (0.02)	1.04 (0.11)	1.07 (0.08)	1.21	0.93 (0.04)	1.08 (0.18)
d	0.09 (0.03)	0.13	0.09 (0.01)	0.09 (0.04)	0.18 (0.09)	0.24	0.13 (0.03)	0.22 (0.14)	0.17 (0.09)	0.43	0.11 (0.12)	0.29 (0.2)

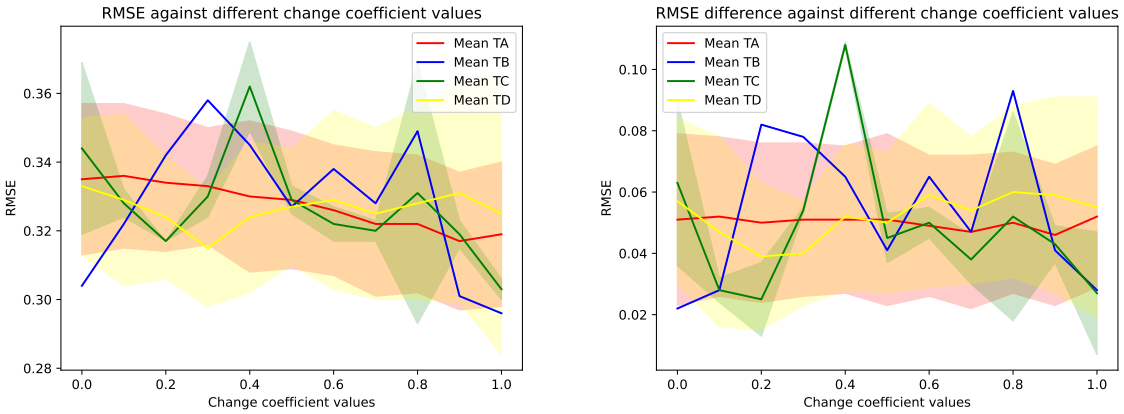
Table 4.4: RMSE results on different scenarios for low data regime. The σ is 0.5 in all settings; c represents the climate change coefficient, h represents the horizon and Δ represents the difference between test and train RMSE



(a) Test RMSE

(b) Difference between train and test RMSE

Figure 4.9: RMSE results on the low data regime against different σ values, $h = 7,\ c = 0.2$



(a) Test RMSE

(b) Difference between train and test RMSE

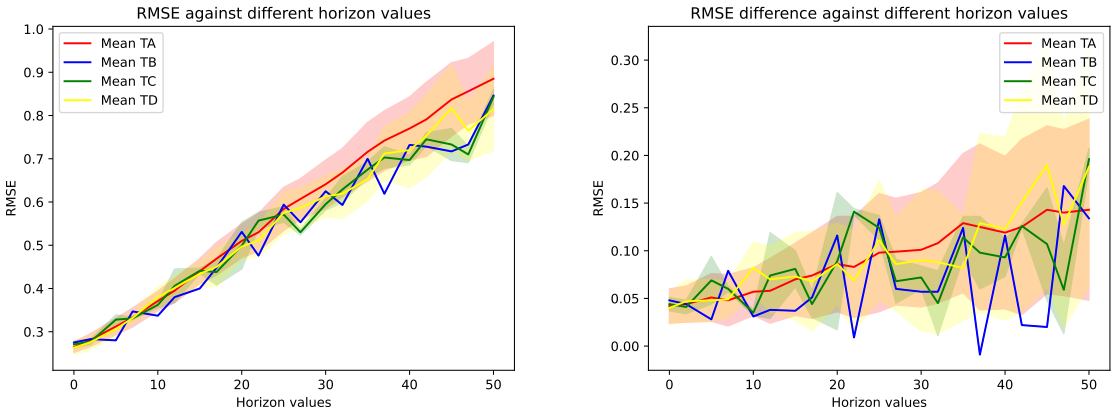
Figure 4.10: RMSE results on the low data regime against different c values, $\sigma = 0.2,\ h = 7$

When discussing the results obtained, several key observations can be made. When using artificial data without horizon and climate change influences, all splitting strategies performed similarly when the STD of the noise was zero. As the STD increased, minor differences began to emerge; however, these differences were minimal, as seen in figure 4.9a. It is important to note that strategies T_B and T_C occasionally exhibited unstable results due to the limited number of data combinations available, with strategies T_A and T_D showing slightly more stability.

In the scenario where only climate change was introduced to the artificial data, the differences between the strategies became more pronounced. Strategies T_B and T_C generally exhibited higher errors for lower STDs, while strategies T_A and T_B tended to have lower errors with strategy T_A being notably

unstable. At an STD of zero, there was no discernible difference between the strategies. As the change coefficients increased, the performance differences among the strategies tended to even out.

When the artificial data incorporated only horizon effects, strategy T_C demonstrated the lowest RMSE, indicating its effectiveness in handling horizon-related issues. Conversely, strategies T_A and T_D showed significant instability. The differences among the strategies became more apparent with larger horizons (30-50), suggesting that strategy T_C is particularly adept at mitigating the challenges posed by horizon effects.



(a) Test RMSE

(b) Difference between train and test RMSE

Figure 4.11: RMSE results on the low data regime against different h values, $\sigma = 0.2,\ c = 0.2$

In the most complex scenario, where both horizon effects and climate change were present, strategies T_A and T_D exhibited the highest RMSE. Although the differences were not highly significant, strategy T_C consistently outperformed the others in terms of RMSE. Overall error rates were high, and the differences between strategies diminished at higher σ. At low σ, all strategies, particularly strategy T_D, showed considerable instability and tendencies towards overfitting.

When looking at the general trend, without considering standard deviation, strategies T_B and T_C are more unstable in the difference between train and test, having higher ups and downs when going, for example, from one horizon value to another one.

Eventually, in the low data regime, where data is sparse, the performance of different data splitting strategies reveals more nuanced differences. Strategy T_C demonstrates robustness in managing horizon effects, meaning it maintains consistent performance across various forecast periods. This robustness is essential for ensuring reliable predictions when data is limited. On the other hand, strategy T_A, despite showing potential in certain scenarios, exhibits instability under specific conditions, as indicated by higher standard deviations.

4.2. Spatial splitting strategies

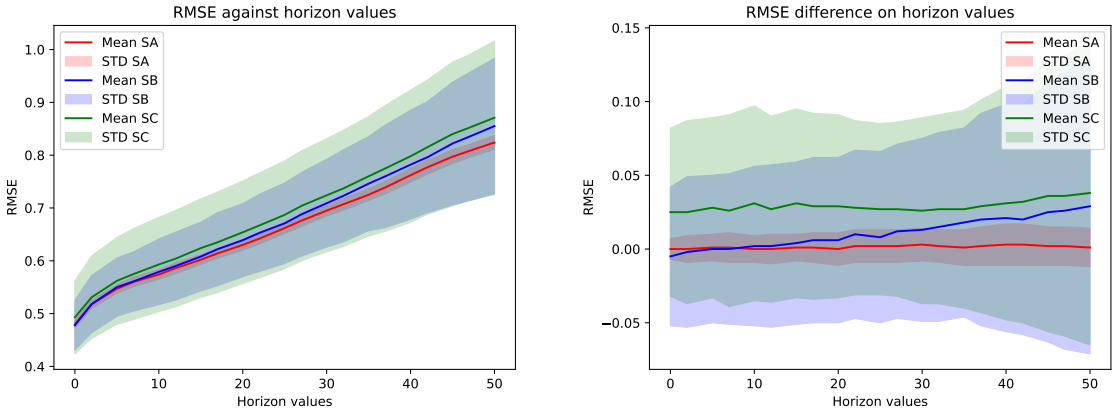
For spatial splitting strategies, only the real data was used, for two reasons: the dataset was already checked when doing the experiments for the temporal strategies and the lack of time. The setting for the spatial strategies was similar to the temporal strategies, therefore the model used was Linear Regression and the performance metric for evaluation was RMSE mean and standard deviation over multiple strategies runs, plus the difference between training and test. Moreover, only 2 scenarios were considered: one where the horizon is nonexistent and one where the horizon takes. The results can be seen in the figures 4.12a and 4.12b and table 4.5. For spatial strategies, 100 samples were taken for strategy S_a, and 4 samples were possible for both S_b and S_c strategies, as only 4 clusters were created by the clustering algorithm.

When changing the horizon, the S_a strategy arises as the most stable and not prone to overfitting, as the differences between the train and test RMSE are close to 0. The other two strategies are performing insignificantly worse, although both the performance and the differences between train and test show

	$h = 0$			$h = 50$		
	S_a	S_b	S_c	S_a	S_b	S_c
Train	0.425 (0.001)	0.427 (0.031)	0.411 (0.034)	0.795 (0.001)	0.798 (0.037)	0.805 (0.053)
Test	0.425 (0.007)	0.425 (0.051)	0.436 (0.078)	0.797 (0.012)	0.827 (0.130)	0.845 (0.148)
Δ	0 (0.008)	-0.002 (0.05)	0.025 (0.064)	0.002 (0.012)	0.029 (0.101)	0.04 (0.104)

Table 4.5: RMSE results on different settings for spatial strategies, $\sigma = 0$; h represents the horizon and Δ represents the difference between test and train RMSE

that these two strategies are highly unstable, as seen in figure 4.12b. This also means that S_b and S_c are prone to overfitting as in one run the mean difference is slightly over 0 and in another run, the mean difference can be around 0.1. One reason for this is that there are not many clusters formed (only 4), and therefore, only 4 runs with different results. This behaviour is more nuanced when the horizon is higher, as the standard deviation of the mean RMSE is rising.



(a) Test RMSE

(b) Difference between train and test RMSE

Figure 4.12: RMSE results against different h values on spatial strategies, $\sigma = 0.2$

Strategy S_b exhibits an interesting behaviour, similar to what was seen on strategies T_B and T_C earlier. When $h = 0$, S_b has a difference between train and test lower than 0, which means that on test data, strategy S_b performs better than on training data, although the standard deviation for the differences is quite large. Same as the temporal strategies, the spatial strategies have a higher RMSE when increasing the horizon values, from above 0.4 when $h = 0$ to approximately 0.8 when $h = 50$, as shown in the figure 4.12a.

In the context of evaluating the robustness of various strategies, strategy S_a demonstrated superior stability, evidenced by the minimal mean RMSE difference between the training and testing datasets. This indicates that strategy S_a consistently performs well on new, unseen data, suggesting that it is less prone to overfitting and can generalize better across different datasets. On the other hand, strategies S_b and S_c, while similar to each other in terms of robustness, exhibited greater instability compared to strategy S_a. This increased instability is reflected in the larger mean RMSE differences between their training and testing datasets, indicating that these strategies may overfit the training data to a greater extent and may not generalize as effectively to new data.

5

Discussion and Conclusion

5.1. Splitting strategies conclusion

Weather prediction has historically relied on complex atmospheric equations, solvable only with significant computational resources. However, the growing availability of meteorological data and advancements in ML has opened new avenues for more efficient weather forecasting. The central problem addressed in this research was seeing a difference in which the data is split and used to train ML models, as the data is auto-correlated.

This thesis explored various data-splitting strategies for weather prediction, focusing on temporal and spatial aspects. Our research aimed to identify the strategies that yield the lowest bias and variance, enhancing the accuracy of the weather forecasts. The thesis evaluated four temporal splitting strategies (T_a, T_b, T_c, and T_d) and three spatial splitting strategies (S_a, S_b and S_c), using real meteorological data from various stations across Europe and generated artificial data (only for testing the temporal splitting strategies). Each strategy was tested incrementally by increasing the horizon values to observe their impact on the RMSE.

The results for real data emphasize that, while strategy T_a shows promise due to its robustness, the risk of overfitting cannot be overlooked. Strategy T_c, though less stable, presents a balanced alternative with its resistance to overfitting. Strategy T_b's high tendency for overfitting and instability limits its applicability. In contrast, Strategy T_d's performance variability based on climatic conditions suggests a need for further contextual analysis when implementing this strategy. These insights underscore the importance of selecting and tailoring forecasting strategies to specific data characteristics and environmental contexts to enhance predictive accuracy and reliability.

The results for artificial data emphasize the critical need to carefully consider data-splitting strategies in both high and low data regimes. The robustness and minimal overfitting of strategy T_c make it a strong candidate in various scenarios, while the stability of strategy T_a is advantageous where consistency is predominant. However, the specific data characteristics and external factors such as horizon effects and climate change must guide the final choice of strategy, ensuring optimal performance and reliability in forecasting tasks.

The interplay between horizon effects and climate change further accentuates these differences. Horizon effects refer to the variability in model performance across different forecasting periods, while climate change introduces additional uncertainty and variability into the data. These factors underscore the importance of selecting an appropriate data splitting strategy that aligns with the specific characteristics of the dataset and the environmental factors at play. The findings highlight the necessity of a context-aware approach to choosing data splitting strategies, ensuring that the selected method is well-suited to the intrinsic properties of the data and the external conditions influencing its behaviour.

Last but not least, the results for spatial splitting strategies show the superior stability of strategy S_a. This makes it a more reliable choice for applications requiring consistent performance across varied

data environments. The spatial analysis included clustering weather stations and evaluating the influence of geographical distribution on prediction performance. We discovered that certain strategies, like clustering and random sampling, can significantly affect the model's ability to generalize across different locations.

The answers to the research questions posed at the beginning of the thesis are the following:

RQ1: "Which temporal data splitting strategy has the lowest bias and lowest variance of the performance in weather forecasting?": This thesis concludes that strategy T_c is particularly effective in handling horizon-related issues and consistently outperforms other strategies in terms of RMSE, making it the preferred choice for temporal data splitting with the lowest bias and variance in weather forecasting performance.

RQ2: "Which spatial data splitting strategy has the lowest bias and lowest variance of the performance in weather forecasting?": This thesis concludes that strategy S_a can be safely recommended for splitting the data, although further research should include a broader range of meteorological features and employ advanced methodologies to capture the intricate dynamics and interactions of these variables for enhanced accuracy and reliability in weather predictions.

5.2. Limitations and new potential directions

Although this work underlines the differences between data splitting strategies, both spatial and temporal, there are some limitations and with those, new directions that will be discussed further.

5.2.1. Validation data

One immediate limitation is the absence of validation data, caused by choosing Linear Regression as an ML model. Linear regression is known to not have hyper-parameters, and hence validation data acted only as test data. Validation data is an important asset in training the ML / DL model, because in the training process, the hyper-parameters need to be trained as well, and the most efficient way to do it is to use some data points as validation data.

This omission of a dedicated validation set is a significant limitation because validation data play a crucial role in the training process of both ML and DL models. Validation data are essential for several reasons:

1. Hyper-parameter Tuning: In ML and DL, models often have numerous hyper-parameters that need fine-tuning to achieve optimal performance. A validation dataset provides a separate set of data to evaluate the model during training, ensuring that adjustments to these hyper-parameters improve generalization rather than overfitting the training data.
2. Early Stopping: Validation data help in implementing early stopping during training. By monitoring the performance on the validation set, training can be halted when the performance starts to decrease, thus preventing overfitting and saving computational resources.
3. Generalization Assessment: Using a validation set allows for an ongoing assessment of the generalisation of the model capability throughout the training process. This ongoing assessment is crucial because it ensures that the performance of the model is not just good on the training data but also on data it has not seen before.

In the study by Schultz et al., validation data were integral because they focused on both ML and DL models, which have complex architectures and numerous hyper-parameters requiring careful tuning [29]. The inclusion of validation data enabled a more robust training process, ensuring the models were well-optimized and capable of generalizing to new data.

Future work should involve exploring other ML models that also necessitate hyperparameter tuning. By including validation data in these future studies, the models can be trained more effectively, leading to more reliable results. Incorporating validation data is not merely a procedural step but a fundamental practice in developing high-performing and generalizable ML and DL models.

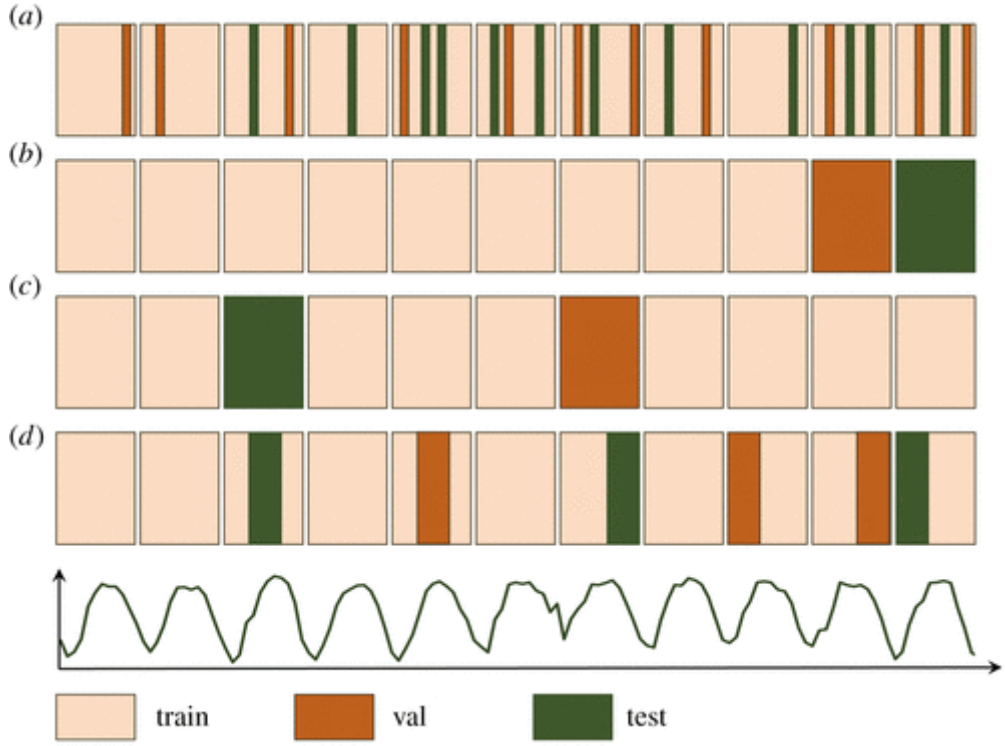


Figure 5.1: Different types of splitting a dataset [29, p. 11]

5.2.2. Other meteorological features

Temperature is widely regarded as one of the most crucial meteorological variables, significantly influencing weather patterns and forming a central element of weather reports globally. Despite its importance, other meteorological factors such as humidity, atmospheric pressure, and precipitation also play vital roles in weather prediction and need to be considered to obtain a comprehensive understanding. For instance, humidity not only affects human comfort levels but also plays a critical role in the formation of precipitation and clouds. Atmospheric pressure is another key indicator, with high pressure typically associated with clear skies and low pressure often leading to stormy weather conditions.

In this thesis, the focus was exclusively on maximum temperature, which inherently limits the scope of the study by excluding other essential meteorological variables. This limitation restricts the ability to capture a more holistic view of weather patterns and their variations. The exclusion of variables like humidity and atmospheric pressure means that the findings of the study are constrained to temperature trends and do not account for the complex interplay between different meteorological factors.

To address this limitation, future studies should consider incorporating a broader range of meteorological features. This would involve employing functions with more impulse responses to analyze the data effectively, as seen in figure 5.2. Impulse response functions, which are typically zero throughout their domain except for a specific value at a certain time, can help in understanding the impact of sudden changes in meteorological variables. However, trends in features such as humidity and atmospheric pressure are often less straightforward than those of temperature, necessitating more complex analytical approaches.

Overall, while the current study provides valuable insights into temperature trends, it underscores the need for a more comprehensive approach that includes other meteorological factors to enhance the accuracy and reliability of weather predictions. Future research should aim to integrate these variables, utilizing advanced methodologies to capture their intricate dynamics and interactions.

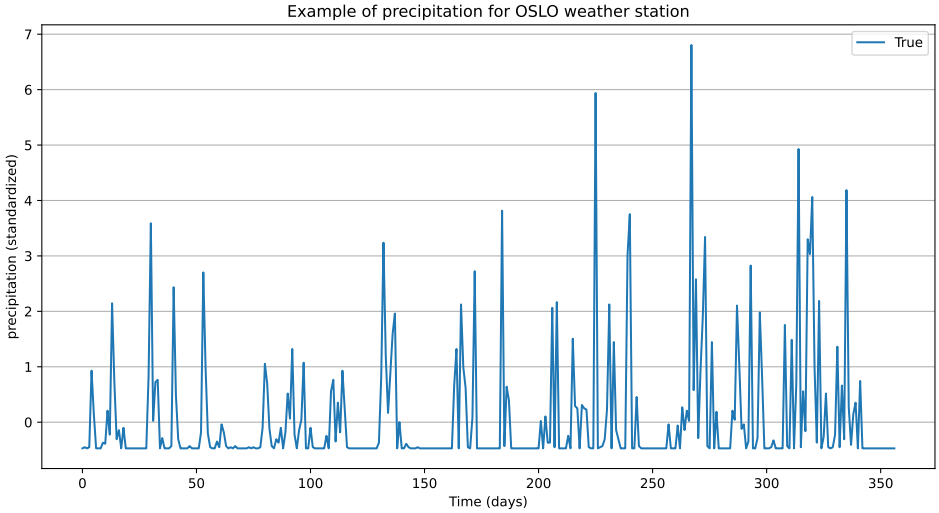


Figure 5.2: Example function of another meteorological feature.

5.2.3. Equal amount of samples between the splitting strategies

In the conducted experiments, it was observed that there existed an unequal distribution of samples across different splitting strategies. This discrepancy was a result of data limitations and the inherent design of the strategies. This unequal distribution led to situations where certain strategies were evaluated more robustly, while others were tested under more constrained conditions. It was recognized that this imbalance in sample distribution could potentially lead to an unfair comparison of the data splitting strategies. Such an unfair comparison could stem from a higher variance for one strategy and a lower variance for another, thereby impacting the reliability of the results. Specifically, a strategy with a larger sample size might show more stable and consistent performance metrics, whereas a strategy with fewer samples might exhibit greater variability and less reliable outcomes.

In consideration of future work, one suggestion is to address this imbalance by equalizing the number of samples for all splitting strategies, both in temporal and spatial contexts. This involves ensuring that each strategy, regardless of its nature, is provided with an equivalent quantity of data points, thereby eliminating the variability caused by sample size differences. By doing so, the comparability and fairness of the evaluation across the different strategies can be enhanced. Additionally, this approach would enable a more accurate assessment of the performance of each strategy by isolating the variable of sample size and focusing on the inherent effectiveness of the strategies themselves. Such methodological improvements are crucial for deriving meaningful insights and making informed decisions based on the experimental results.

Another proposed future work is to increase the number of samples used for the splitting strategies. This expansion would provide a more robust dataset for each strategy, reducing the impact of outliers and enhancing the statistical power of the comparisons. By having a larger pool of data, the experiments would yield more reliable and generalizable results, allowing for a more thorough evaluation of each performance of the strategy under varying conditions. This increase in sample size would help ensure that the observed outcomes are more reflective of the true capabilities of the strategies, thus contributing to the overall validity and credibility of the research findings.

5.2.4. New data splitting strategies

An important area for further research is the development of alternative data splitting strategies. One promising direction is to create a spatial equivalent of the temporal data splitting strategy T_d. For instance, a spatial data splitting approach could involve forming clusters of weather stations and selecting one station from each cluster for the testing set, while the remaining stations in the cluster comprise the training set. This approach would mirror the temporal splitting method, thereby providing a comprehensive analogy between temporal and spatial strategies. Implementing and evaluating this method

would be a significant step toward enhancing the robustness and generalizability of weather prediction models.

Additionally, future work could explore the combination of temporal and spatial data splitting strategies. For example, researchers could select random samples from all weather stations within a cluster for the test set and compare the RMSE of this approach against the traditional method of using the last year of data as the test set. This combined strategy might leverage the strengths of both temporal and spatial splitting, potentially leading to more accurate and reliable models.

Another avenue for future research is to identify and implement entirely new types of data splitting strategies. One such strategy could involve considering the climatic conditions of the weather stations. By categorizing stations based on their climate type and then splitting the data accordingly, models could be trained and tested in a manner that accounts for climatic variability. This approach might offer more nuanced insights and improve model performance in diverse climatic conditions.

Overall, these proposed strategies highlight the potential for innovative data splitting methods to enhance the accuracy and generalizability of predictive models in meteorology. Further research in this area could lead to significant advancements in weather forecasting and related fields.

5.2.5. Sliding window values

A notable limitation of this thesis is the consistent use of a default value of 7 for all sliding window parameters in the conducted experiments, except for the horizon. This uniform parameter setting may not fully capture the variability and potential effectiveness of different parameter combinations in various contexts.

Given this limitation, a compelling direction for future research would be to systematically evaluate the impact of varying the sliding window parameters: the window size (i), the output (o), and the step size (s). By exploring combinations where these parameters differ from the default value of 7 and from each other (for instance, $i = 3$, $o = 1$, $s = 2$), it would be possible to gain deeper insights into how these different configurations influence the performance and outcomes of the splitting strategies. This approach could reveal optimal parameter settings for specific applications and enhance the generalizability of the findings.

Investigating these alternative parameter values could involve a series of controlled experiments designed to isolate the effects of each parameter individually as well as in combination. Such studies could utilize performance metrics relevant to the application domain, such as prediction accuracy, computational efficiency, and the robustness of data splitting strategies. Additionally, comparing the results across both real and artificial datasets would be valuable to understand the consistency of these effects.

Ultimately, this line of inquiry holds promise for advancing the methodology by identifying more versatile and effective parameter settings for sliding window approaches. By addressing this limitation, future research can build on the current findings and find the differences in RMSE between different scenarios in parameter values in sliding windows.

5.2.6. Other datasets

One notable limitation of this thesis project was the potential lack of diversity in the datasets used. Although the chosen dataset proved to be reliable, evidenced by the good overlay of the results with the artificial data, the validity and robustness of the splitting strategies could benefit from being tested on a wider variety of datasets. Specifically, the current conclusions of this study are drawn from a single dataset, which may limit the generalizability of the findings to other contexts or climates.

To address this limitation, future research should consider applying the proposed splitting strategies to additional datasets. For example, exploring weather datasets from different geographical regions and climates could provide a more comprehensive understanding of the effectiveness of the splitting strategies. Such datasets might include those capturing tropical, arid, temperate, and polar climates, each presenting unique characteristics and challenges.

Furthermore, another avenue for future work involves training the models on one dataset while testing them on another. This approach would more accurately simulate real-world conditions where the training data and operational data may not always come from the same source or possess the same

statistical properties. By doing so, we can evaluate the ability of the models to generalize and adapt to unseen data, thus enhancing their practical applicability in diverse scenarios.

Overall, expanding the range of datasets and incorporating cross-dataset validation would not only mitigate the limitations identified but also strengthen the evidence for the robustness and flexibility of the splitting strategies. These steps are crucial for advancing the research and ensuring that the developed methods can be reliably applied in various real-world situations.

5.2.7. Graph distance threshold

In our experimental setup, we empirically set the distance threshold between cities to 500 km. While this threshold was chosen based on initial observations and practical considerations, it lacks a rigorous scientific justification. This introduces a limitation to our approach, as the chosen threshold might not be optimal for all scenarios or datasets. To address this, future research should focus on developing and implementing robust evaluation metrics that can accurately measure the performance of clustering algorithms under varying distance thresholds. Such metrics would provide a more scientific basis for determining the most appropriate distance threshold, enhancing the validity and generalizability of the clustering results.

Furthermore, the utilization of synthetic data can greatly aid in this effort. By generating synthetic datasets that simulate the distribution of weather stations across a geographical area, researchers can systematically analyze how different distance thresholds affect clustering performance. These artificial datasets would allow for controlled experiments where the true distances between points are known in advance. Consequently, this approach enables the precise evaluation of clustering methods, providing clear insights into their efficacy and guiding the selection of optimal distance thresholds. Incorporating synthetic data into the evaluation process could lead to more robust and adaptable clustering techniques, applicable to diverse geographical and meteorological contexts.

5.3. Final remarks

This thesis assessed various data splitting strategies in NWP. Despite some limitations, our findings offer valuable insights that can be applied to other fields, such as medical diagnostics and financial forecasting. These strategies can enhance the accuracy and reliability of predictive models, highlighting the importance of effective data splitting across different domains. This work lays a foundation for future research to improve predictive capabilities in various disciplines.

References

[1] Llinet Benavides Cesar et al. “Review on spatio-temporal solar forecasting methods driven by in situ measurements or their combination with satellite and numerical weather prediction (NWP) estimates”. In: *Energies* 15.12 (2022), p. 4341.

[2] Vincent D Blondel et al. “Fast unfolding of communities in large networks”. In: *Journal of statistical mechanics: theory and experiment* 2008.10 (2008), P10008.

[3] Bogdan Bochenek and Zbigniew Ustrnul. “Machine learning in weather prediction and climate analyses—applications and perspectives”. In: *Atmosphere* 13.2 (2022), p. 180.

[4] Dongjin Cho et al. “Comparative assessment of various machine learning-based bias correction methods for numerical weather prediction model forecasts of extreme air temperatures in urban areas”. In: *Earth and Space Science* 7.4 (2020), e2019EA000740.

[5] Antonio S Cofıno et al. “Bayesian networks for probabilistic weather prediction”. In: *15th Eureopean Conference on Artificial Intelligence (ECAI)*. Citeseer. 2002.

[6] Julian J Faraway. “Does data splitting improve prediction?” In: *Statistics and computing* 26 (2016), pp. 49–60.

[7] Vladimir Agafonkin. *Fast geodesic approximations with Cheap Ruler*. 2016. URL: `https://blog.mapbox.com/fast-geodesic-approximations-with-cheap-ruler-106f229ad016`.

[8] Naveen Goutham et al. “Using machine-learning methods to improve surface wind speed from the outputs of a numerical weather prediction model”. In: *Boundary-Layer Meteorology* 179 (2021), pp. 133–161.

[9] Dorothy K Hall et al. “Comparison of satellite-derived and in-situ observations of ice and snow surface temperatures over Greenland”. In: *Remote Sensing of Environment* 112.10 (2008), pp. 3739–3749.

[10] Mark Holmstrom, Dylan Liu, and Christopher Vo. “Machine learning applied to weather forecasting”. In: *Meteorol. Appl* 10 (2016), pp. 1–5.

[11] Florian Huber et al. *Weather prediction dataset*. Version v5. Jan. 2023. DOI: `10.5281/zenodo.7525955`. URL: `https://doi.org/10.5281/zenodo.7525955`.

[12] RG Isaacs, RN Hoffman, and LD Kaplan. “Satellite remote sensing of meteorological parameters for global numerical weather prediction”. In: *Reviews of Geophysics* 24.4 (1986), pp. 701–743.

[13] AHM Jakaria, Md Mosharaf Hossain, and Mohammad Ashiqur Rahman. “Smart weather forecasting using machine learning: a case study in tennessee”. In: *arXiv preprint arXiv:2008.10789* (2020).

[14] *Weather Prediction*. URL: `https://www.kaggle.com/datasets/thedevastator/weather-prediction`.

[15] Diksha Khiatani and Udayan Ghose. “Weather forecasting using hidden Markov model”. In: *2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*. IEEE. 2017, pp. 220–225.

[16] Loi Lei Lai et al. “Intelligent weather forecast”. In: *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826)*. Vol. 7. IEEE. 2004, pp. 4216–4221.

[17] Remi Lam et al. “Learning skillful medium-range global weather forecasting”. In: *Science* 382.6677 (2023), pp. 1416–1421.

[18] Blake LeBaron and Andreas S Weigend. “A bootstrap evaluation of the effect of data splitting on financial time series”. In: *IEEE transactions on neural networks* 9.1 (1998), pp. 213–220.

[19] Yang Li et al. “Super-resolution probabilistic rain prediction from satellite data using 3d u-nets and earthformers”. In: *arXiv preprint arXiv:2212.02998* (2022).

[20] Zaiqiao Meng et al. “Exploring data splitting strategies for the evaluation of recommendation models”. In: *Proceedings of the 14th acm conference on recommender systems*. 2020, pp. 681–686.

[21] *Will climate change make weather forecasting less accurate?* 2023. URL: https://climate.mit.edu/ask-mit/will-climate-change-make-weather-forecasting-less-accurate.

[22] Azam Moosavi, Vishwas Rao, and Adrian Sandu. “Machine learning based algorithms for uncertainty quantification in numerical weather prediction models”. In: *Journal of Computational Science* 50 (2021), p. 101295.

[23] Sravani Nalluri, Somula Ramasubbareddy, and G Kannayaram. “Weather prediction using clustering strategies in machine learning”. In: *Journal of Computational and Theoretical Nanoscience* 16.5-6 (2019), pp. 1977–1981.

[24] Aye Nandar. “Bayesian network probability model for weather prediction”. In: *2009 International Conference on the Current Trends in Information Technology (CTIT)*. IEEE. 2009, pp. 1–5.

[25] Assaf Rabinowicz and Saharon Rosset. “Cross-validation for correlated data”. In: *Journal of the American Statistical Association* 117.538 (2022), pp. 718–731.

[26] Markus Reichstein et al. “Deep learning and process understanding for data-driven Earth system science”. In: *Nature* 566.7743 (2019), pp. 195–204.

[27] David R Roberts et al. “Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure”. In: *Ecography* 40.8 (2017), pp. 913–929.

[28] Sebastian Scher and Gabriele Messori. “Predicting weather forecast uncertainty with machine learning”. In: *Quarterly Journal of the Royal Meteorological Society* 144.717 (2018), pp. 2830–2841.

[29] MG Schultz et al. “Can deep learning beat numerical weather prediction?, Philos”. In: *Roy. Soc. A*. Vol. 379. 2194. 2021, pp. 10–1098.

[30] Robert H Shumway, David S Stoffer, and David S Stoffer. *Time series analysis and its applications*. Vol. 3. Springer, 2000.

[31] Josie Garthwate. *Climate of chaos: Stanford researchers show why heat may make weather less predictable*. 2021. URL: https://news.stanford.edu/2021/12/14/warming-makes-weather-less-predictable/#:~:text=For%20precipitation%2C%20predictability%20falls%20by,5%20C%20increase%20in%20temperature..

[32] *STATE OF PLAY: UNDERSTANDING IN SITU DATA*. URL: https://insitu.copernicus.eu/state-of-play/understanding-in-situ-data.

[33] Xiaoni Wang and Catherine Prigent. “Comparisons of diurnal variations of land surface temperatures from numerical weather prediction analyses, infrared satellite estimates and in situ measurements”. In: *Remote Sensing* 12.3 (2020), p. 583.

[34] Jeremy Deaton. *Climate change could make weather harder to predict*. 2022. URL: https://www.washingtonpost.com/weather/2022/01/25/climate-change-weather-unpredictable/.

A

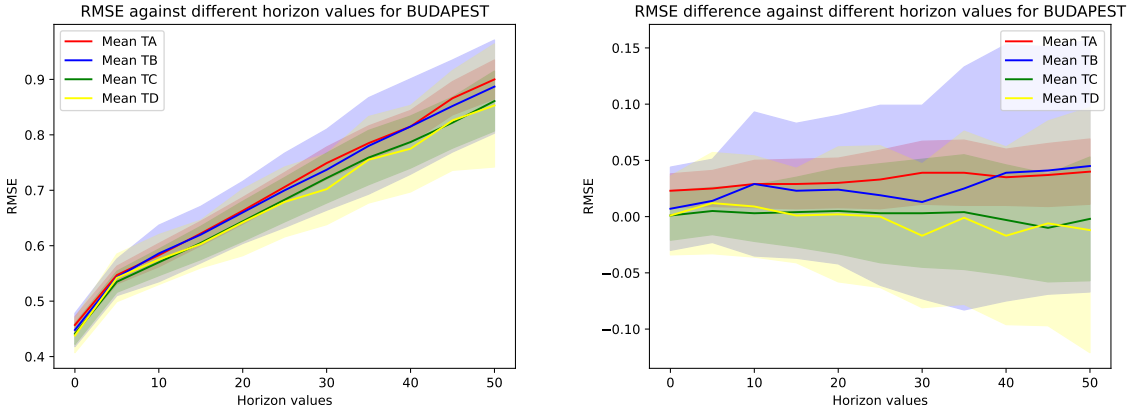
Further results and graphs

	Budapest weather station								Heathrow weather station							
	T_A		T_B		T_C		T_D		T_A		T_B		T_C		T_D	
	$h=0$	$h=50$	$h=0$	$h=50$	$h=0$	$h=50$	$h=0$	$h=50$	$h=0$	$h=50$	$h=0$	$h=50$	$h=0$	$h=50$	$h=0$	$h=50$
Train	0.44 (0.01)	0.87 (0.02)	0.44 (0.01)	0.84 (0.03)	0.44 (0.01)	0.86 (0.01)	0.44 (0.01)	0.87 (0.01)	0.51 (0.01)	0.88 (0.01)	0.51 (0.01)	0.88 (0.01)	0.51 (0.01)	0.88 (0.01)	0.51 (0.01)	0.88 (0.01)
Test	0.46 (0.02)	0.90 (0.04)	0.45 (0.03)	0.89 (0.08)	0.44 (0.02)	0.86 (0.05)	0.44 (0.03)	0.87 (0.11)	0.53 (0.02)	0.91 (0.04)	0.55 (0.06)	0.93 (0.11)	0.53 (0.05)	0.9 (0.09)	0.53 (0.04)	0.9 (0.1)
Δ	0.02 (0.02)	0.04 (0.03)	0.01 (0.04)	0.05 (0.11)	0 (0.02)	0 (0.06)	0 (0.03)	0 (0.11)	0.02 (0.02)	0.04 (0.03)	0.04 (0.07)	0.05 (0.12)	0.02 (0.05)	0.02 (0.09)	0.02 (0.04)	0.02 (0.11)

Table A.1: RMSE results on different scenarios for Budapest and Heathrow stations. h represents the horizon and Δ represents the difference between train and test RMSE

	Perpignan weather station							
	T_A		T_B		T_C		T_D	
	$h=0$	$h=50$	$h=0$	$h=50$	$h=0$	$h=50$	$h=0$	$h=50$
Train	0.48 (0.01)	0.86 (0.02)	0.49 (0.02)	0.87 (0.02)	0.49 (0.01)	0.87 (0.01)	0.49 (0.01)	0.87 (0.01)
Test	0.5 (0.02)	0.89 (0.04)	0.53 (0.06)	0.93 (0.09)	0.5 (0.04)	0.87 (0.1)	0.51 (0.03)	0.88 (0.11)
Δ	0.02 (0.02)	0.04 (0.03)	0.04 (0.05)	0.06 (0.1)	0.01 (0.04)	0 (0.1)	0.02 (0.04)	0.01 (0.11)

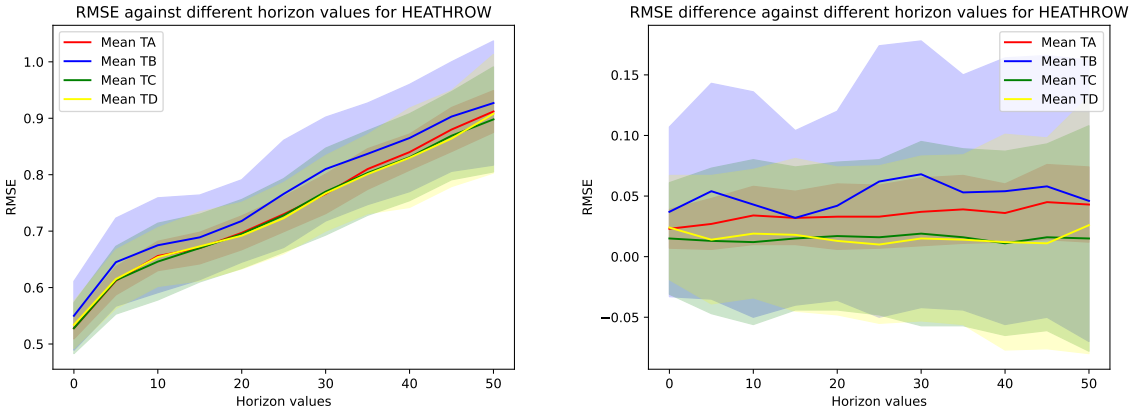
Table A.2: RMSE results on different scenarios for Perpignan station. h represents the horizon and Δ represents the difference between train and test RMSE



(a) Test RMSE

(b) Difference between train and test RMSE

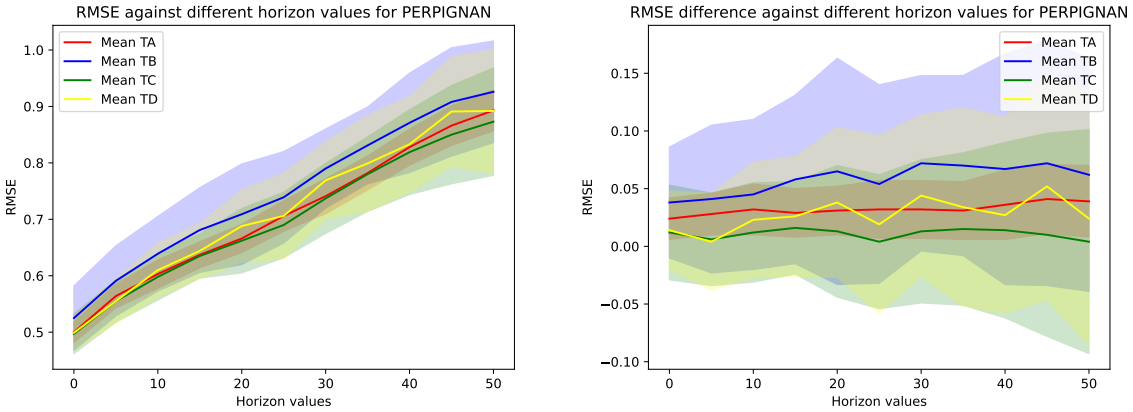
Figure A.1: RMSE results against different h values for Budapest station on maximum temperature feature



(a) Test RMSE

(b) Difference between train and test RMSE

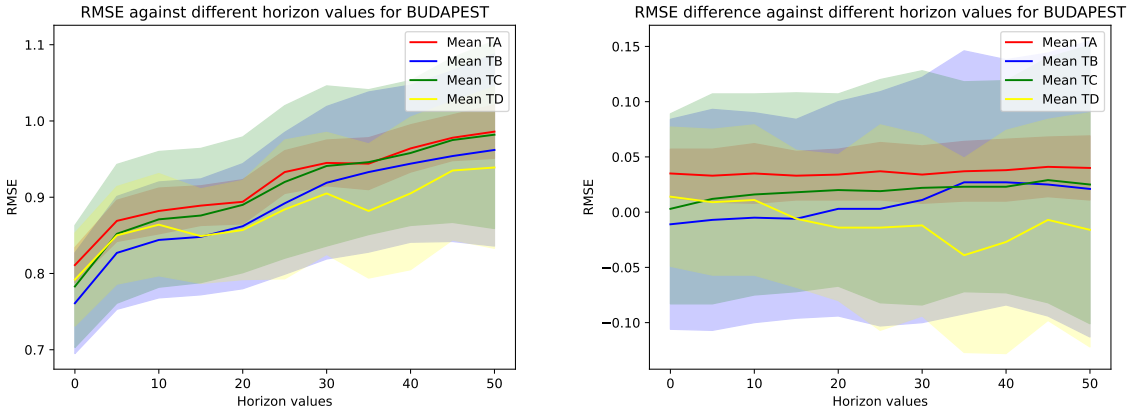
Figure A.2: RMSE results against different h values for Heathrow station on maximum temperature feature



(a) Test RMSE

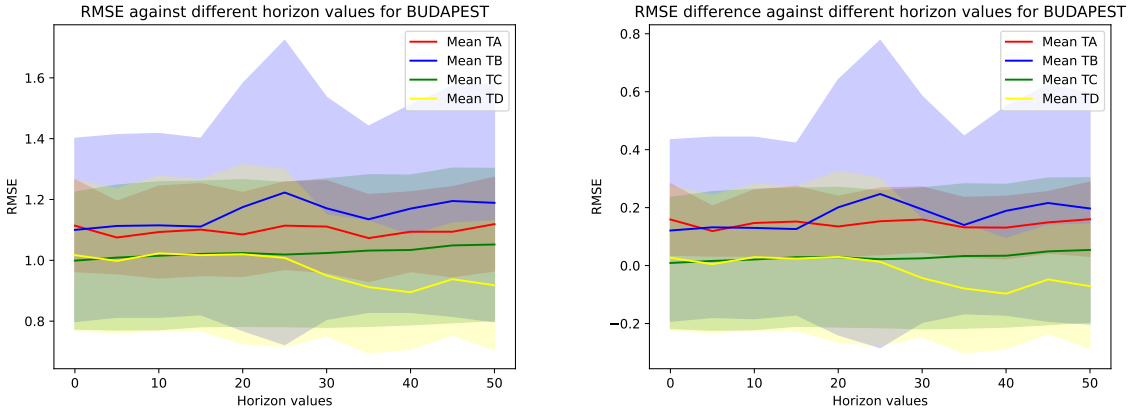
(b) Difference between train and test RMSE

Figure A.3: RMSE results against different h values for Perpignan station on maximum temperature feature



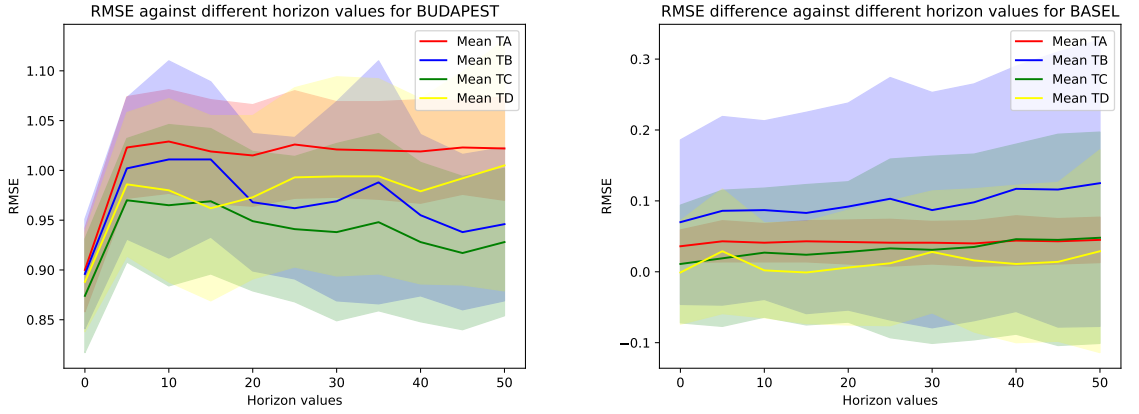
(a) Test RMSE **(b)** Difference between train and test RMSE

Figure A.4: RMSE results against different h values for Budapest station on humidity feature



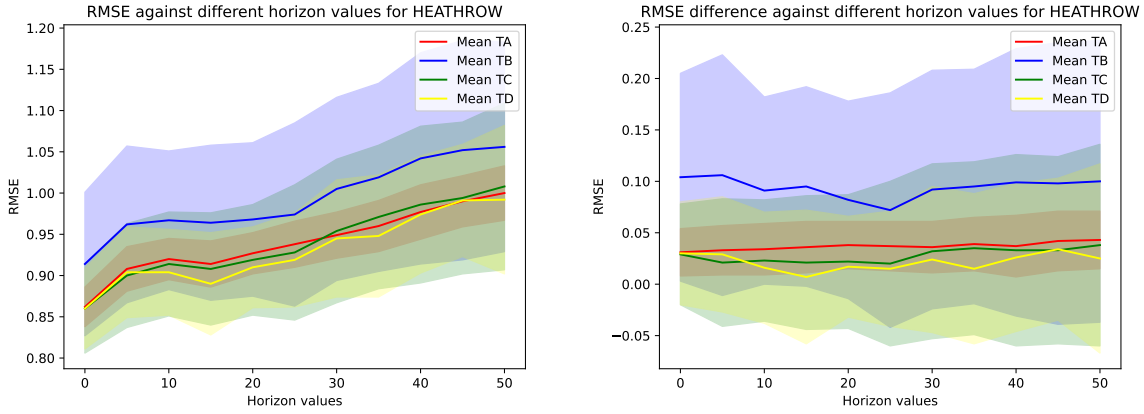
(a) Test RMSE **(b)** Difference between train and test RMSE

Figure A.5: RMSE results against different h values for Budapest station on precipitation feature



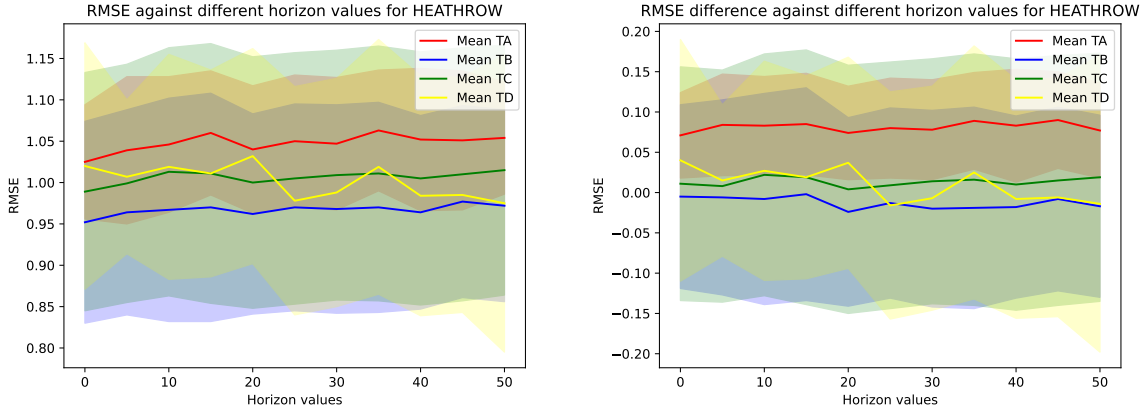
(a) Test RMSE **(b)** Difference between train and test RMSE

Figure A.6: RMSE results against different h values for Budapest station on atmospheric pressure feature



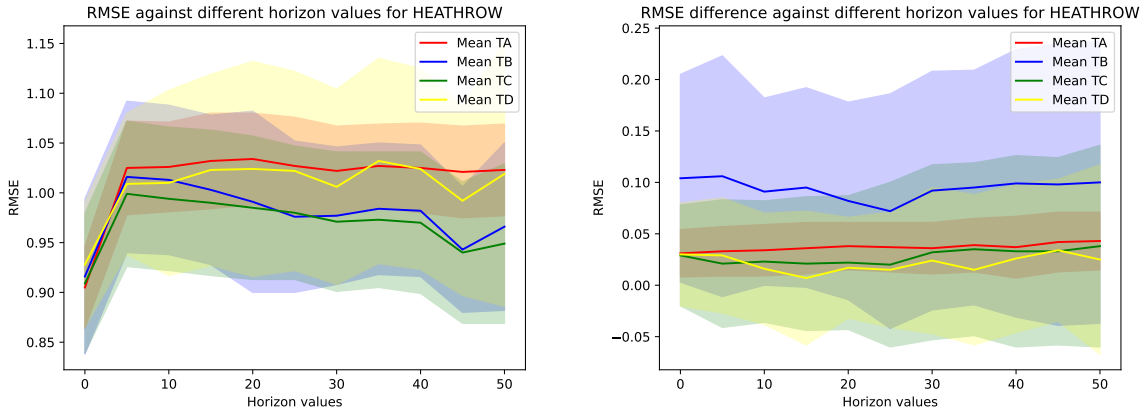
(a) Test RMSE **(b)** Difference between train and test RMSE

Figure A.7: RMSE results against different h values for Heathrow station on humidity feature



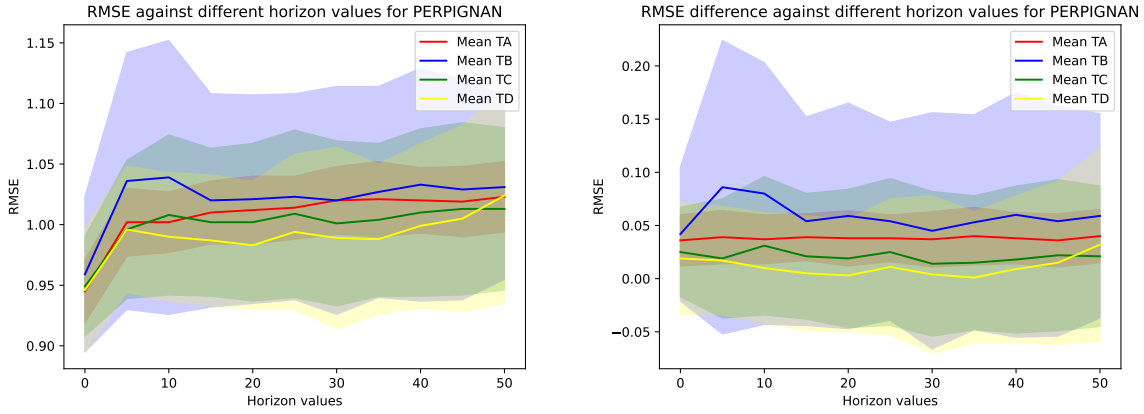
(a) Test RMSE **(b)** Difference between train and test RMSE

Figure A.8: RMSE results against different h values for Heathrow station on precipitation feature



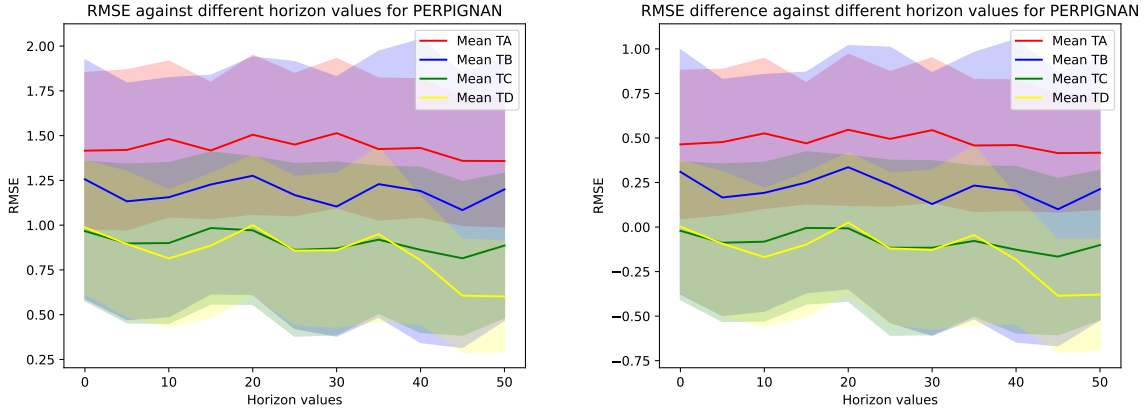
(a) Test RMSE **(b)** Difference between train and test RMSE

Figure A.9: RMSE results against different h values for Heathrow station on atmospheric pressure feature



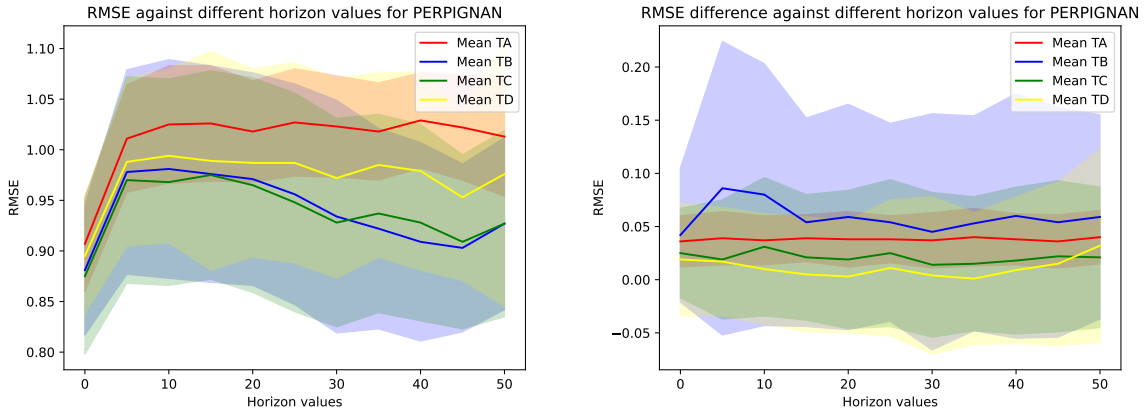
(a) Test RMSE **(b)** Difference between train and test RMSE

Figure A.10: RMSE results against different h values for Perpignan station on humidity feature



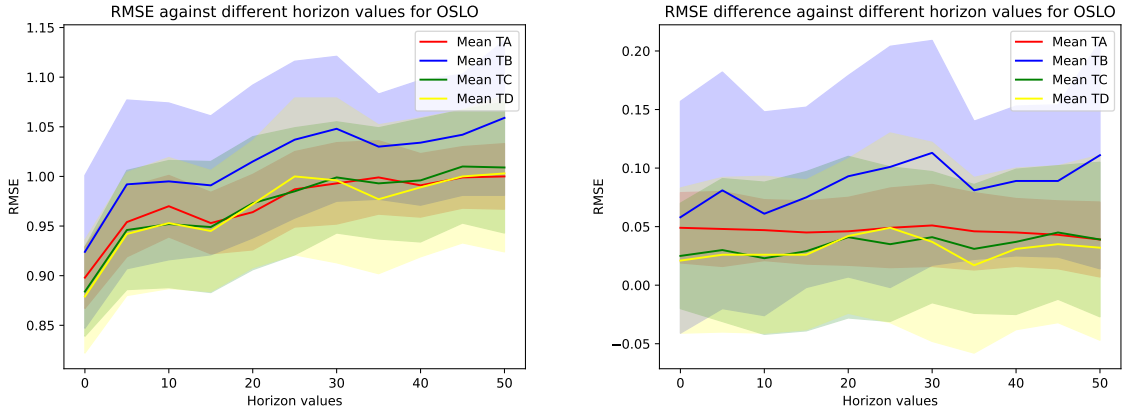
(a) Test RMSE **(b)** Difference between train and test RMSE

Figure A.11: RMSE results against different h values for Perpignan station on precipitation feature



(a) Test RMSE **(b)** Difference between train and test RMSE

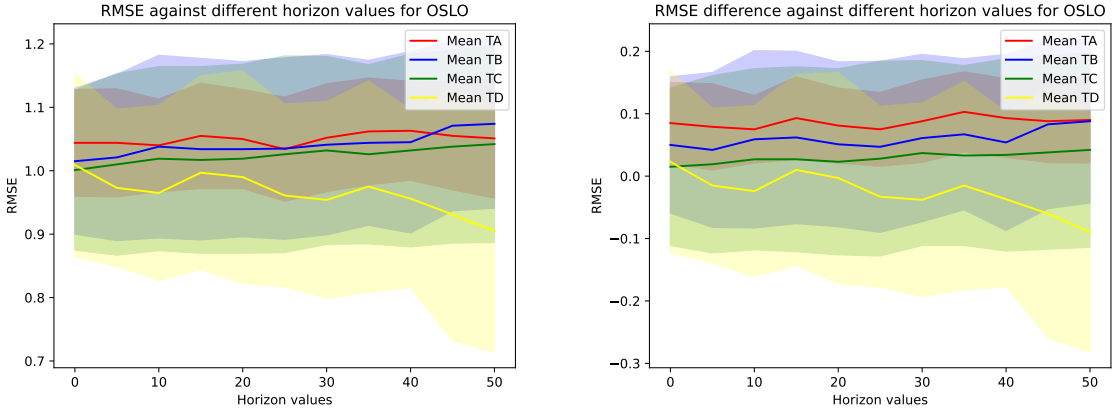
Figure A.12: RMSE results against different h values for Perpignan station on atmospheric pressure feature



(a) Test RMSE

(b) Difference between train and test RMSE

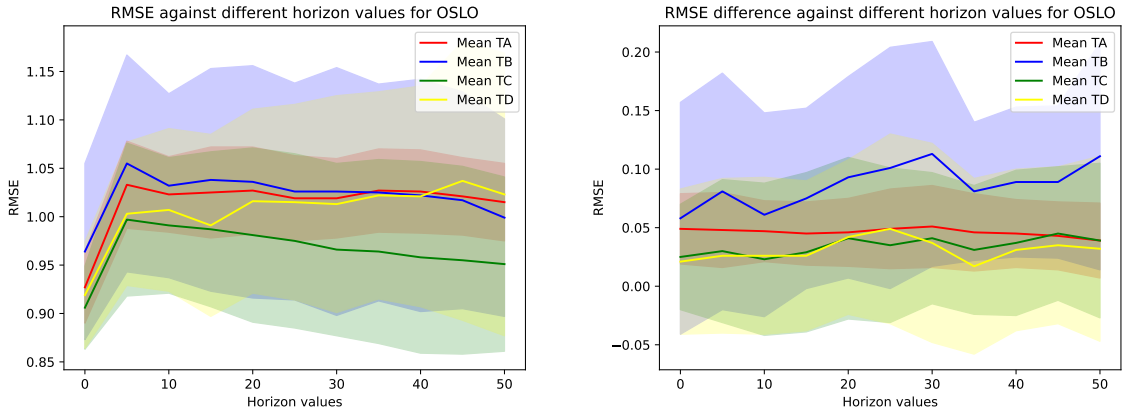
Figure A.13: RMSE results against different h values for Oslo station on humidity feature



(a) Test RMSE

(b) Difference between train and test RMSE

Figure A.14: RMSE results against different h values for Oslo station on precipitation feature



(a) Test RMSE

(b) Difference between train and test RMSE

Figure A.15: RMSE results against different h values for Oslo station on atmospheric pressure feature