German
Research Center
for Artificial
Intelligence

Universität
Bremen

Ambient Air Pollutants Prediction

Enhancing Established Regional Forecasting Systems for Urban Environments using Machine Learning

MASTER THESIS
at the German Research Center for Artificial Intelligence

Department of Mathematics and Computer Science
Faculty of Science & Technology
University of Bremen

presented by
Hannes Gelbhardt
Course of studies: Computer Science
Registration number: [redacted]

on
13th December, 2023

Examiner: Prof. Dr. Dr. h.c. Frank Kirchner

Supervisor: Dr. Nicolás Navarro-Guerrero

Abstract

Accurately predicting air pollutant concentration remains challenging but essential in preventing the public from being exposed to high concentrations that lead to several hundred thousand premature deaths across Europe yearly (European Environment Agency, 2023).

This research focuses on enhancing hourly regional pollutant forecasts in local urban environments using machine learning (ML). To achieve this, a data set was collected, harmonized, and preprocessed to represent eleven major German cities that served as the basis for answering the research question. The ability to predict $PM_{2.5}$ and NO_2 at a target location with and without corresponding local measurements at that point was evaluated for several employed ML algorithms. Incorporating measurements at the designated sites enabled the locally implemented ML algorithms to diminish the error in the regional forecast by 29.77% and 44.99% for $PM_{2.5}$ and NO_2, respectively. Even in the absence of measurements, a notable reduction in error by 15.14% and 18.71% compared to the regional forecast was evident at the target location.

Thus, the presented study shows how valuable locally employed ML algorithms can be to enhance regional forecasts in urban environments, suggesting that these proposed methods can help warn the citizens about estimated high concentrations if operational.

Zusammenfassung

Das akkurate Vorhersagen von Luftschadstoffen bleibt eine anspruchsvolle aber wichtige Aufgabe zur Vermeidung der Exposition von Menschen gegenüber hohen Schadstoffkonzentrationen, die jährlich zu mehreren Hunderttausend vorzeitigen Todesfällen in Europa führen (European Environment Agency, 2023).

Die vorliegende Studie fokussiert sich auf die Verbesserung von regional Vorhersagen der stündlichen Schadstoffkonzentration im lokalen, städtisch geprägten Raum mithilfe von maschinellem Lernen (ML). Hierfür wurde ein Datensatz akquiriert, harmonisiert und vorverarbeitet, um elf große Städte in Deutschland zu repräsentieren, welche als Grundlage zur Beantwortung der Forschungsfrage dienten. Auf diesem Datensatz wurde die Fähigkeit verschiedener ML-Algorithmen für die Vorhersage von $PM_{2.5}$ und NO_2 an einem bestimmten Punkt mit und ohne den dortigen Messungen evaluiert. Die Einbeziehung von Messungen an den festgelegten Standorten ermöglichte es den lokal implementierten ML-Algorithmen, den Fehler in der regionalen Vorhersage um 29,77% bzw. 44,99% für $PM_{2.5}$ und NO_2 zu reduzieren. Selbst in Abwesenheit von Messungen war eine bemerkenswerte Fehlerreduktion um 15,14% bzw. 18,71% im Vergleich zur regionalen Vorhersage am Zielort evident.

Die vorliegende Studie verdeutlicht den Wert lokal eingesetzter ML-Algorithmen für die Verbesserung regionaler Vorhersagen im urbanen Umfeld und legt hierdurch nahe, dass die vorgeschlagenen Methoden, falls operationell, dazu beitragen, Bürger besser vor hohen Luftschadstoffkonzentrationen warnen zu können.

Contents

List of Figures

List of Tables

1 Chapter Introduction

According to the World Health Organization, 2022, air pollution is defined as any biological, chemical, or physical agent that contaminates indoor or outdoor environments by modifying the attributes of the atmosphere. The five most harmful air pollutants for human health and the environment named by the World Health Organization (WHO) are particulate matter (PM), ground-level ozone (O_3), carbon monoxide (CO), nitrogen dioxide (NO_2), and sulfur dioxide (SO_2). The particular matter is an inhalable mixture of different particles, which can be further subdivided into particulate matter with a diameter $< 10\mu g/m^3$ (PM_{10}) and fine particulate matter with a diameter $< 2.5\mu g/m^3$ ($PM_{2.5}$). Primarily, the latter is known to have a severe impact on human health, resulting in 238.000 premature deaths in 2020 among the population of the 27 European Union Member states, making it the most significant individual environmental health hazard in the European Union (European Environment Agency, 2023). Familiar sources of air pollutants are motor vehicles, industrial processes, forest fires, and volcanic activities. One crucial factor for all pollutants is the concentration in the air, which can be measured using ground-level sensors or estimated via satellite imagery. Predicting future pollutant levels could lower the risk of exposure to high concentrations of these pollutants or take measures to reduce the concentration. The danger from exposure can be divided into short- and long-term exposures. While the air pollutant concentration has to be considerably higher for short-term exposure to be harmful, already relatively low pollutant concentration can have a similar effect on a long-term basis (World Health Organization et al., 2021).

The prediction of future air pollutants can roughly be divided into chemical transport models (CTMs) that aims to model the atmospheric chemistry considering a specific pollutant species and statistical models that learn from experience to estimate the pollutant concentrations. While CTMs are also applied to regional scenarios (Marécal et al., 2015), they are typically used on a larger scale by modeling the advection (the movement of bulk) and diffusion (the movement or dispersion inside the bulk) of the particles. Ground-level observations measure the in-situ concentrations and are more suitable for capturing local trends. Often, a high temporal resolution and more accurate measurements make the latter

Table 1.1: The table shows the ranges for the air quality indices. All values correspond to a daily average pollutant concentration of $\mu g/m^3$. From top to bottom, the AQI defined by the EEA is Good, Moderate, Poor, Very Poor, and Extremely Poor. For each index, a suggestion of how to behave is given.

AQI	$PM_{2.5}$	PM_{10}	NO_2	O_3	SO_2
Good	0-10	0-20	0-40	0-50	0-100
Fair	10-20	20-40	40-90	50-100	100-200
Moderate	20-25	40-50	90-120	100-130	200-350
Poor	25-50	50-100	120-230	130-240	350-500
Very Poor	50-75	100-150	230-340	240-380	500-750
Extremely Poor	> 75	> 150	> 340	> 380	> 750

particularly interesting for the domain of statistical models or machine learning (ML). Due to the complex nature of CTMs and the spatial relatively coarse prediction resolution, the field of model output statistic (MOS) has been adapted from weather forecasting systems to improve the prediction performance in local environments by incorporating local measurements to correct the systematic bias at the specific location.

Challenges in predicting exposures include the precise forecast of multiple time steps ahead, e.g., hours for the short term and months or years for the long term at a specific location. Another challenge is the forecast of an episode that represents a sudden peak or drop in pollutant concentration, which rarely occurs. Various protective measures can be taken depending on the short- or long-term exposure scenarios. An immediate reduction in pollutant concentration can be achieved if, for example, the traffic is redirected from a polluted area or a nearby power plant is shut down (Boznar et al., 1993). Additionally, the population in the area can be warned in advance of a rising pollutant concentration, giving them enough time to leave or avoid the area. Long-term exposure estimation, for example, can be helpful in urban planning so that new residential buildings are not placed into areas that are estimated to be (or become) highly polluted (Frenkiel, 1956).

Table 1.1 shows the exposure risks to human health defined by the European Environmental Agency (EEA) for the five primary pollutants. It additionally orders the expected danger of the different pollutant species decreasingly from left to right. While the table gives a good intuition of the impact of exposure on the public, the WHO even defines lower values as more harmful. Recent studies analyzed the correlation between pollutant concentration and premature death in low-exposure environments. They concluded that even $PM_{2.5}$ exposure below 3 $\mu g/m^3$ increases the risk of premature deaths (Brauer et al., 2022). Since the risks of air pollutants to human health have already been known for many decades, the history of air pollutant modeling dates back at least to the 1950s. In the following, a short history of air pollutant modeling (and related literature) is given.

Frenkiel, 1956 analyzed the influence of different pollutant sources on the mean main pollutant level in a 16-square-mile grid over Los Angeles County. The pollutants are defined by the emission source, including two industrial plants, the traffic density per grid, and the estimated number of private incinerators per grid. A mathematical model is constructed that considers different meteorological factors and estimates that the long-term development of air pollutant concentration nearly doubled from 1954 to 1980. Boettger and Smith, 1961 classified the SO_2 and PM concentration into four intensity levels for the next day, using a rule-based system on the meteorological data. The highest accuracy is achieved for the winter season with 63% for SO_2 and 60% accuracy for the PM concentration. Three years later, Clarke, 1964 implemented a diffusion model based on Gaussian distribution to estimate the NO_x and SO_2 concentration at the city center by incorporating the estimated pollutant sources of the neighboring area, and meteorological factors. They showed that they could successfully predict the pollutant concentration at the center.

The term MOS was first coined by Glahn and Lowry, 1972 in the domain of weather forecast. The authors applied multiple linear regression using local weather measurements as predictors (e.g., maximum temperature for the current day) to improve the prediction of regional weather forecasting systems (e.g., maximum temperature) over the east united states of America (USA) by reducing the systematic bias at the particular location. Motivated by the success of MOS, Klein and Glahn, 1974 extended the approach and made it operational over the entire USA by applying one multiple linear regression model for every station, two seasons, and each objective. While Clarke, 1964 neglected the historical air pollutant concentration at a particular target station, McCollister and Wilson, 1975 implemented a stochastic linear model that learned from the daily maximum or the past 24 hourly average pollutant concentrations to predict the following daily maximum or hourly average concentrations, respectively. The result outperformed the forecast by human weather experts by a small margin while considering fewer variables (e.g., meteorological factors).

To address the challenge of forecasting an episode (a rapid increase followed by a rapid decrease of the concentration at the target location), Fronza et al., 1979 incorporated an advection-diffusion model extended with the Kalman prediction for the real-time forecasting of SO_2 concentration. As a result, while considering different meteorological factors, the researchers accurately identified high pollutant concentration episodes. One of the first implementations of artificial neural networks (ANNs) to predict pollutant concentration at different locations was investigated by Boznar et al., 1993. The research focused on predicting the SO_2 concentration at different target sites around a thermal power plant 30 minutes ahead. Another research of neural networks was conducted by Comrie, 1997. A comparison between ANNs and other regression models for 1-hour daily maximum O_3 forecasting was analyzed here. Additionally, the inputs included meteorological factors and the previous day's maximum O_3 concentration. Contrary to the researchers' expectations, the ANN only slightly outperformed the multivariate regression models. Another comparison by Gardner and Dorling, 1999 showed that their implementation of the ANN outperformed the more traditional linear regression models in predicting the hourly NO_x and NO_2 concentration for London in all experiments. The suggestion that more complex models like neural networks outperform linear regression models was also underlined two years later by Elkamel et al., 2001.

One of the earliest research that applied MOS to improve CTMs in the urban area of Paris was conducted by Blond et al., 2003. The authors tested different setups with and without using the CTM prediction as an additional predictor. They successfully applied a kriging method (Cressie, 1993) to interpolate spatial dependencies of pollutant concentration from local observation by incorporating the forecast of the CTM, resulting in an estimate of the error field of the CTM that can be used to correct the systematic bias over the region and improve the prediction performance of O_3 in that area. They concluded that incorporating the CTM prediction into the model provided valuable information, especially in sparse ground-level measurement networks.

Aldrin and Haff, 2005 implemented generalized additive models to forecast $PM_{2.5}$, PM_{10}, NO_2 and NO_x separately and included additionally the number of passing motorised vehicles at the measurement sites into their model. Yildirim and Bayramoglu, 2006 predicted the daily pollutant concentration of SO_2 and PM by utilizing an adaptive neuro-fuzzy model and achieved an root mean squared error (RMSE) of 30 $\mu g/m^3$ for PM for a winter season. Wilczak et al., 2006 used MOS to improve the performance of a multi-model air quality forecast ensemble that predicted O_3 in North America. They employed a simple 7-day running mean bias correction for each station and predicted each hour of the day, showing that this achieves the highest performance of individual and ensemble forecasts. They additionally note that the highest bias correction of the individual hours is achieved during the past day. Another implementation of MOS was performed by Honoré et al., 2008, which examined an operational forecasting system of O_3 in France. They estimated the bias of the prediction at each site for every hour to interpolate it over the spatial dimension by using the kriging method described in Blond et al., 2003. An implementation of an Elman network for predicting the next hourly pollutant concentration was incorporated by Prakash et al., 2011. The researcher preprocessed the data using a wavelet-transform and reported an mean absolute error (MAE) of 6.52 $\mu g/m^3$ for the next hours $PM_{2.5}$ prediction. Feng et al., 2015 investigated another wavelet transformation-based approach. The authors additionally included and transformed the air mass trajectories to predict the daily average $PM_{2.5}$ concentration two days in advance at multiple locations. They achieved an average daily MAE of 12.31 $\mu g/m^3$ $PM_{2.5}$.

Today, CTMs, local statistical models, and MOS are still successfully applied to estimate future pollutant concentration. For example, the established air pollution forecast model for Europe is based on a median ensemble of different CTMs, predicting each day the next 96 hours for different pollutants at a spatial resolution of 0.1° lati- and longitude (corresponding to approximately 10 km^2) for a grid over Europe (Marécal et al., 2015). The hourly forecast includes ten different pollutant concentrations, and the performance between the individual models varies in different scenarios. Since the computational cost is relatively high for today's standard, the predictions are available approximately 8 hours after the calculation starts. Apart from forecasting a gridded area, predicting the future pollutant concentration at a specific station is performed extensively throughout literature and used by the EEA to inform the public about future pollution concentration levels (EEA, 2023). Even though some of the recent research combines the regional forecast with in-situ measurements over Europe by applying ML algorithms (Bertrand et al., 2023), the performance of this combined method has not yet been assessed for multiple urban environments in Germany. This research is going to answer this question from different perspectives.

1.1. Aim and Objectives

This research aims to evaluate the performance of local and regional approaches to improve regional forecasts and find the most suitable algorithms to forecast the next 23 hours of target pollutants in multiple German urban environments.

To achieve this, comparable literature is reviewed to identify information relevant to the research aim, including expectations on the underlying data set, data preparation, the algorithms for the prediction, and how to measure and compare the success of the different predictions. For comparison, two suitable pollutants that serve as targets are additionally chosen. In the next step, the gained knowledge are used to collect and prepare a suitable data set and implement feasible local ML algorithms identified during the literature review. Since the collected data set diverges from the ones proposed in the literature, the hyper parameters (HPs) of each identified ML algorithm is optimized. To assess the performance of the different ML algorithms, three different scenarios are evaluated. First, similar to Bertrand et al., 2023, the prediction at a specific target station is examined. Here, the regional forecast is compared against the local prediction. Second, it is evaluated if this forecast can be improved by incorporating the measurements of neighboring in-situ stations. The third scenario answers the question of whether the prediction of the target pollutant can be improved if only neighboring stations are included without utilizing the historical in-situ observations at the target location. How to fuse the predictions of the neighboring stations is also be evaluated. All three scenarios are performed with and without the regional forecast to assess the influence.

1.2. Contribution of the Work

Even though MOS has already been applied in the context of air pollutant forecast (Bertrand et al., 2023; Blond et al., 2003; Honoré et al., 2008; Wilczak et al., 2006), none of the researchers evaluated ML to improve the regional forecast in urban environments, more particularly in several major German cities. While Blond et al., 2003 and Honoré et al., 2008 model the error or systematic bias of the regional forecast of O_3 by inter- or extrapolating the error over the spatial dimension using kriging, these researches do not utilize ML to populate the error surface. They might be unable to capture the pollutant concentrations more affected by local factors. Wilczak et al., 2006 relied on a simple bias correction of the regional forecast across the USA by calculating the running mean error of the seven days, also without applying ML algorithms and not particularly for urban environments. Bertrand et al., 2023, on the other hand, recently applied MOS using ML, at hundreds of sites across Europe. However, this research does not evaluate the improvement of the regional forecast in a local urban environment, nor does it consider the spatial dependencies between the stations.

Since most people in Europe live in cities, for example, 77.7% of the population in Germany (destatis, 2023), knowledge about how to accurately predict air pollutants can be performed in urban environments is particularly important. Therefore, this study aims to fill this gap across multiple major cities in Germany. If the research yields promising results, the identified methods could be implemented across cities to automatically improve the regional forecasts locally, enhancing the short-term warning capability for the public.

Chapter 2 Literature Review

The following chapter presents past and current research, outlining the state-of-the-art for air pollutant prediction. The section aims to categorize the relevant information from the revised articles into the steps that must be implemented to answer the overall research question. Because of its crucial importance to the successful implementation, the data sources the different studies build on are presented in Section 2.1. The available data is often preprocessed or preselected to enable or simplify the learning process for the different machine learning (ML) algorithms, and therefore presented in the following Section 2.2 and Section 2.3, respectively. When the data is ready, it can be fed to the proposed algorithm so that the different vital contributions of the revised studies are presented in Section 2.4. The performance of the trained ML algorithms can be evaluated in multiple ways. A revision of how several authors approached this challenge can be found in Section 2.5.

2.1. Data sets

This section presents the various input data the researchers employed in their studies. In addition, information about the temporal and spatial resolution is provided, as well as the time span of the data and supplementary information.

Ground-level pollutant measurement stations are the essential source for predicting future exposure levels. They measure the accumulated pollutant concentration of different harmful gases, primarily measured in micrograms per cubic meter (μg/m^3) or parts per billion (ppb). Given the molecular weight of the pollutant, the temperature, and the air pressure, the units can be converted interchangeably, except for particulate matter (PM), because the composition of different molecules varies and might, therefore, be unknown. In addition, meteorological factors like wind speed, rainfall, and temperature highly influence the distribution and dispersion of the different pollutant concentrations. Table 2.3 gives an overview of the most common input variables used to predict future air pollutants in the different

Table 2.1: Overview of the different input variables. While all authors used $PM_{2.5}$ and most meteorological data, some incorporated the other pollutant concentrations and few NO_2 and NO_x

Article	Meteorological data	$PM_{2.5}$	PM_{10}	CO	NO	NO_2	NO_x	O_3	SO_2
Zheng et al., 2015	✓	✓	✓	✓		✓		✓	✓
X. Li et al., 2017	✓	✓							
Kleine Deters et al., 2017	✓	✓							
Biancofiore et al., 2017	✓	✓	✓	✓					
Huang and Kuo, 2018	✓	✓							
Liang et al., 2018	✓	✓	✓	✓		✓		✓	✓
Tao et al., 2019	✓	✓							
Zhao et al., 2019		✓	✓	✓		✓		✓	✓
Du et al., 2019		✓	✓	✓		✓		✓	✓
Qiao et al., 2019		✓							
Qin et al., 2019	✓	✓							
Zhou et al., 2019	✓	✓	✓	✓	✓	✓	✓	✓	✓
Castelli et al., 2020	✓	✓		✓		✓		✓	✓
Chang et al., 2020	✓	✓	✓	✓	✓	✓	✓	✓	✓
Zhang et al., 2021		✓							
Zeng et al., 2022	✓	✓							
Jin et al., 2022	✓	✓							
Akbal and Ünlü, 2022	✓	✓	✓	✓	✓	✓	✓		✓
Saez and Barceló, 2022	✓	✓	✓			✓		✓	
Tian et al., 2022	✓	✓	✓	✓		✓			✓

studies.

Besides the input variables shown in Table 2.1, the authors included additional features in their data set. One example is to incorporate time-related features, e.g., the hour of the day, the day of the week, or the month of the year (Castelli et al., 2020; Kleine Deters et al., 2017; Liang et al., 2018; Zhao et al., 2019). These features enable the models to learn recurring patterns in the data. Other studies included the pollutant concentrations of neighboring stations to predict the target stations' future pollutant concentration (Jin et al., 2022). Incorporating the prediction of chemical transport models (CTMs) (more specifically, the ensemble forecast of Copernicus Atmospheric Monitoring Service (CAMS)) to improve the local predictions is evaluated by Bertrand et al., 2023.

The number of analyzed ground-level stations dramatically varies from a single station (e.g., Castelli et al., 2020) to multiple thousand stations (e.g., Zheng et al., 2015). Also, the underlying data sets' time range varies from half a year (Castelli et al., 2020) to 10 years (Saez and Barceló, 2022). The sampling frequency of the

Table 2.2: Overview of the different satellite-based input variables. Displayed are the MF, the AOD, the NDVI and the CTM data.

Article	MF		NDVI	CTM
Hu et al., 2017	✓	✓		✓
T. Li et al., 2017	✓	✓	✓	
Di et al., 2019	✓	✓	✓	✓
X. Meng et al., 2021	✓	✓	✓	
Zamani et al., 2019		✓		
Muthukumar et al., 2021		✓		

ground-level pollutant concentration measurements is mostly hourly and in some cases daily (Biancofiore et al., 2017; Kleine Deters et al., 2017).

Apart from ground level measurements, it is also possible to estimate ground pollutant concentration (in particular fine particulate matter with a diameter $< 2.5\mu g/m^3$ ($PM_{2.5}$)) from satellite images. For example, the most commonly used aerosol optical depth (AOD) measurements are recorded with the Moderate Resolution Imaging Spectroradiometer of the Earth Observing System. The measurement products most relevant in estimating surface $PM_{2.5}$ concentrations are 470nm and 550nm (Di et al., 2019). The data availability of satellites is sparse and influenced by multiple factors. First, one orbit of a corresponding satellite takes approximately two days, depending on the satellite's distance to Earth. Additionally, factors that cause missing data are, for example, cloud cover, snow, and high uncertainty (Di et al., 2019). Therefore, some authors use CTM to interpolate the missing values. As additional inputs, other satellite images, e.g., the meteorological fields (MF) or the normalized difference vegetation index (NDVI), a measure of the amount of vegetation for remote sensing, are employed. Table 2.2 gives an overview of data sets that estimate the ground-level pollutant concentration over an area.

Multiple authors also included various land use variables for each estimated grid cell. The most common were local road cover, forest cover, and population density. The grid cells' spatial resolution varied from $1km^2$ (Di et al., 2019) to $12km^2$ (Hu et al., 2017). The period for the underlying data set ranged from one year (Hu et al., 2017) to 15 years (Di et al., 2019). Since the prediction of surface pollutant concentration can be verified via comparison with ground-level stations, many researchers include them in their data set. The number of ground-level stations significantly differs from under 20 (Muthukumar et al., 2021) to over 2000 stations (Di et al., 2019).

In this section, a variety of possible predictors were presented. They can mainly be categorized into historical pollutant concentration measurements, meteorological factors, and supplementary information, such as time features, population density, and elevation. The choice of the data set and the utilized input variables are crucial for predicting air pollutant concentration. The size of the data set additionally plays an important role. It has been shown that while some researchers build their research on a single pollutant time series, others rely on multiple pollutants and ground-level measuring stations. In particular, the studies incorporating AOD data to predict ground-level pollutant concentration utilized more ground-level stations in their research. Also, the time interval in which the data set was recorded differs from six months to decades. The next section presents an overview of different preprocessing techniques to modify the data.

2.2. Data preprocessing

Raw input data often requires additional preprocessing before it can be handled meaningfully or more efficiently by learning algorithms. One example of air pollutant prediction is categorical meteorological variables like names (e.g., "cloudy"). Another example is that, for some learning algorithms, all numerical input variables should lie on the same scale so that they are not weighed differently according to their magnitude. Various other preprocessing methods exist, some of which are presented below. There are different approaches to handle missing data in the time series. In comparison, some authors imputed the missing time steps of ground-level sensors by linear interpolation (Akbal and Ünlü, 2022; Jin et al., 2022; Zhao et al., 2019), and others used the mean value (Tian et al., 2022) and the previous valid value (Tao et al., 2019). Additionally, dropping the sample (Kleine Deters et al., 2017), using the 2^{nd} order polynomial (Castelli et al., 2020) or applying the Akima smooth curve supplement method were used to interpolate missing values (Chang et al., 2020). To impute the AOD data Di et al., 2019, trained a random forest (RF) to predict the AOD value of a grid cell by using all other predictors from the data set as input variables. The grid cells where the AOD data was present served as ground truth. A tropospheric chemistry-driven model was applied by Hu et al., 2017 to impute the missing AOD data. Many algorithms require numerical input, whereas categorical features are often non-numeric. Additionally, encoding categorical features ensures that the model can effectively learn patterns and relationships within the data, improving the algorithm's performance. In this sense, one of the simplest method to transform categorical features is to apply the one-hot encoding (X. Li et al., 2017; Liang et al., 2018; Zheng et al., 2015). Other authors assigned an integer value to the different categories (Tao et al., 2019), which might lead to a false representation if the underlying feature is not ordinal. Castelli et al., 2020 showed one way to overcome this. The authors transformed the time-related features using the sine and cosine (e.g., $\cos(2\pi\times\text{hour}/24)$) to reflect a repeating pattern. Kleine Deters et al., 2017 calculated the sine and cosine from the wind direction, which was transformed from polar to Cartesian coordinates and multiplied by the wind speed.

Different normalization and standardization techniques were performed to transform the other input features on a similar scale. The input features were normalized between either 0 and 1 or -1 and 1 (Huang and Kuo, 2018; Jin et al., 2022; X. Li et al., 2017; Qiao et al., 2019; Tian et al., 2022) by some authors and standardized by others (Chang et al., 2020; Tao et al., 2019; Zhou et al., 2019). In some cases, outliers were dropped from the data set (Castelli et al., 2020; Kleine Deters et al., 2017). For instance, Castelli et al., 2020 identified the most relevant time lags (past time steps) via auto-correlation. The authors additionally applied the Yeo-Johnson transformation to convert the data and make it more robust against abnormal observations in the data set. Another transformation was performed by Zeng et al., 2022. To capture different patterns in the data, they decomposed the signal into six different sub-signals by calculating the extended stationary wavelet transform on the input signal. To capture long- and short-term exposure Saez and Barceló, 2022 binned the data set into long-term exposure (monthly average) and short-term exposure (daily average). Akbal and Ünlü, 2022, on the other hand, averaged all stations in one city to represent the overall pollutant level.

Zheng et al., 2015 implemented a more complex preprocessing approach. Based on a specific target station, the neighboring stations were accumulated in grid cells corresponding to direction and distance. For example, in eight directions, the closest considered circle is 30km, and the furthest is 300km away. Grid cells that do not include monitoring stations were not considered. The pollutant levels and meteorological factors were averaged for all other cells over stations in them. If the ground-level measurements were provided hourly, all authors modeling ground level $PM_{2.5}$ concentration from AOD data are averaging the hourly data at least to the daily mean. Hu et al., 2017 additionally averaged the meteorological satellite data over the lower tropospheric layers up to 20km per grid. Furthermore, they used a convolutional neural network (CNN) model to weigh the spatial relationship of neighboring grid cells and used the output as an additional predictor. T. Li et al., 2017 performed another approach of weighing neighboring cells. The authors used a distance measure to restrict the network's input to neighboring cells or time steps with decreasing influence that falls inside a fixed threshold. This section presented research covering different approaches to handling the input data. Although some of the most common techniques for imputing or normalizing the data have been identified, the specific preprocessing steps often depend on the use case. Overall, the preprocessing steps of the data sets containing AOD and other satellite images were more complex.

2.3. Selection of input variables

Selecting suitable input variables can reduce the amount of the input dimension to simplify the prediction problem. One of the simplest ways of choosing the correct predictors in literature is through literature references (Huang and Kuo, 2018; Zhao et al., 2019; Zhou et al., 2019). While X. Li et al., 2017 performed a correlation analysis between the $PM_{2.5}$ values of 12 different stations to justify the spatial relationship between them, Castelli et al., 2020; Mao et al., 2021; Tao et al., 2019 applied a correlation analysis by using a covariance matrix between the different predictors and the feature importance. Akbal and Ünlü, 2022 utilized another feature importance technique. The authors used extra trees and random forests to perform a backward selection to rank the input features and considered the first 15 as input for the models. A simple distance in kilometers was used as a threshold to include $PM_{2.5}$ and particulate matter with a diameter $< 10\mu g/m^3$ (PM_{10}) of neighboring stations with a shorter distance than 50km as predictors. Stations placed near an industrial area are added as input ad-hoc, even though the distance is further than 50km away.

The authors of the revised literature provided rarely information on how the input variables were chosen, either because no selection method was performed or because it was not stated. Nevertheless, reducing the input dimension and selecting relevant predictors is essential to air pollutant prediction.

2.4. Proposed model

This section includes the different algorithms to handle the preprocessed data and predict future air pollutant concentrations. It focuses on how temporal and spatial dimensions are merged and how this information is used to predict future air pollutants. Starting from articles that investigate the prediction of pollutants from one station, studies that consider the spatial relationship between stations are revised, followed by studies that model the spatial grid over an area from satellite data. A comparison among a linear regression model, a feed-forward neural network, and an Elman network to predict the average $PM_{2.5}$ concentration of up to three days given the current time step was made by Biancofiore et al., 2017. Zhou et al., 2019 utilized the Kendall Tau algorithm to rank the time lag (past time steps) of neighboring stations to serve as input for a target location and predict the $PM_{2.5}$ value up to 4 hours ahead using a support vector machine (SVM). Another SVM was applied by Castelli et al., 2020 to separately predict different air pollutant concentrations. Using model output statistic (MOS) Bertrand et al., 2023 trained multiple ML algorithms including the linear regression, ridge regression, Lasso, RF and gradient boosting regressor (GBR) to predict $PM_{2.5}$, PM_{10}, nitrogen dioxide

(NO_2) and ozone (O_3) for a mean hourly or daily concentration at specific target sites.

Akbal and Ünlü, 2022; Mao et al., 2021; Zeng et al., 2022; Zhang et al., 2021, all implemented a variation of the long-short term memory (LSTM) network to predict future $PM_{2.5}$ concentration. While Zhang et al., 2021 compared the model with and without first decomposing the $PM_{2.5}$ using the empirical mode decomposition, Mao et al., 2021 proposed a sliding LSTM that takes the prediction of the last time step as input to the current. Finally, Akbal and Ünlü, 2022 compared the proposed LSTM with feed-forward and convolutional layer combinations. To model spatial and temporal relationships, Zheng et al., 2015 implemented an ensemble consisting of four different methods. First, a linear regressor that modeled the local trend from the past pollutant measurements and weather-related predictors of the current time step to predict the change of $PM_{2.5}$ concentration at the target time step. The second component used an artificial neural network (ANN) to capture the global trend for a specific target station. To combine the local and global trends, a decision tree regressor was trained to weigh both outputs dynamically for the local weather variables. Since sudden drops in pollutant concentration were rarely observed and not often reflected in the training data, the fourth component was derived from the training data to recognize these events by defining simple mathematical rules applied to the weather data. Each model was trained for different cities and future time slots. For each of the first 6 hours, one ensemble (considering local and global factors) was trained. For time ranges up to 25-48h, individual models for the minimal and maximal $PM_{2.5}$ concentration were trained separately.

Given the meteorological data as input, Kleine Deters et al., 2017 predicted the current day's average $PM_{2.5}$ concentration. The performance of multiple models was compared, including a Convolutional Generalization Model (CGM) that incorporated the spatial relationship based on the wind speed and wind direction. To predict the $PM_{2.5}$ concentration of the next time steps, X. Li et al., 2017 used an LSTM to extract the time-dependent features of all air pollutant stations simultaneously. Together with the meteorological data and the time features, the output of the LSTM was fed into a fully connected layer that predicted the air pollutant concentration of the desired time step for each of the 12 stations. The authors trained individual models for each time step and combined the results to predict multiple time steps in the future. Zhao et al., 2019 implemented a very similar approach. The authors trained a separate LSTM for every 36 Stations in Beijing. Afterward, a fully connected layered model was used to combine the predictions of a station of interest and its four closest ones to forecast the target locations' $PM_{2.5}$ concentration of the following real-valued 1-6 hours and the minimal and maximal $PM_{2.5}$ for the intervals 7-12, 13-24, 25-48 respectively.

Huang and Kuo, 2018 utilized a Conv1D-Net to extract the features per station and an LSTM to handle the temporal relationship between the extracted features. Similarly, Du et al., 2019 and Tao et al., 2019 trained a 1D-ConvNet for every station that extracted features for the current time step. The output of each ConvNet was then concatenated into one representation and fed into a bidirectional Gated Recurrent Unit (GRU) network to learn the temporal dependencies. Qin et al., 2019 modeled a broader spatial relationship. In there research, a CNN was trained to convolve the spatial relationship between the 14 Stations of the target and neighboring cities into the prediction of the $PM_{2.5}$ μg/m^3 concentration. The predictions of each time step were then fed into an LSTM model that handled the temporal dimension and predicted the final result for the target city. Chang et al., 2020 also included stations that were further away by incorporating the different data sources of the areas into the LSTM with three different processing streams for the local, the neighboring, and the abroad data set, which merged the output prediction of the individual streams into one prediction simultaneously during training. Another LSTM was utilized by Qiao et al., 2019 to predict the next time step. The authors additionally decomposed the $PM_{2.5}$ signal into different high and low-frequency bins. Before the decomposition was fed to the LSTM, a stacked auto-encoder (SAE) was trained to reduce the dimensionality of each signal decomposition. The prediction of the different decompositions was then denormalized and reconstructed into one forecast value. Tian et al., 2022 performed another unsupervised pretraining. The authors trained a deep belief network (DBN) to predict future time steps. The training of the DBN can be divided into two stages. The model was built layer after layer in the first stage, starting with the input layer. Each layer consisted of a restricted Boltzmann machine (RBM) trained to reproduce the input at the output while the dimension from input to output was decreased. Next, new layers were stacked to the previously trained RBM to construct the DBN consisting of multiple RBMs. In the second stage, the previously initiated model was trained to predict future time step, acting as a normal multi-layer perceptron.

While Liang et al., 2018 modeled the spatial and temporal relationship through a multi-level attention network, Jin et al., 2022 proposed an algorithm that can be divided into two parts. The first part correlated the different stations over the spatial dimension by considering the correlation and redundancy. In particular, the maximal information coefficient and a distance entropy algorithm were used. Since the algorithm was parametric, the authors used a Bayesian optimization approach to find the most suitable parameters to correlate the stations. The variables with high correlation and low redundancy were then selected as input to the second part of the proposed algorithm, which consisted of a variational Bayesian GRU network. This network differed from standard GRUs regarding how the weights were represented. In contrast to using fixed learned weights, each network weight was represented by a learned probability distribution from which new weights were sampled during inference. Saez and Barceló, 2022 proposed an algorithm that incorporated all 143 stations in the spatial area and learned the relationship between them. The stochastic partial differential equation was used to find a Gaussian Random Field (GRF) "with local neighborhood and sparse precision matrix [...] that best represented the Matern field." A generalized linear mixed model (GLMM) was used to predict each target value. More precisely, two GLMMs were initiated and trained per pollutant (one for long- and one for short-term). To include randomness that can model seasonal variances, the integrated nested Laplace approximations were used to predict the pollutants based on the output of the GRF.

The following researchers additionally used satellite images to improve the prediction of ground-level pollutant concentration. For this purpose, Muthukumar et al., 2021 implemented a graph CNN to learn the spatial correlation between weather data. The graph was trained through self-supervision, e.g., nodes (stations) of the graph were randomly hidden during the training to create video-like weather data with higher resolution. The learned output served as input to a Conv-LSTM combined with the air pollutants of the ground-level stations and the remote sensing data. Zamani et al., 2019 followed the question of whether satellite data can improve the ground-level prediction of $PM_{2.5}$. They used ground-level historical $PM_{2.5}$ and meteorological data to combine them with and without aerosol satellite images. To predict the $PM_{2.5}$ of the next time step, they incorporated the RF and extreme gradient boosting (XGB) models and highlighted the feature importance. They showed that the AOD data did not positively influence the prediction performance. They argued that this could be related to the high amount of missing values for this predictor (94%).

Subsequently, studies that aim to estimate the ground-level $PM_{2.5}$ from AOD data and other predictors are revised. For example, Hu et al., 2017 estimated the ground level $PM_{2.5}$ concentration of several grid cells from AOD data using an RF. Another RF was implemented by X. Meng et al., 2021. This study aimed to fill the spatial grid cells of the study area with estimations of the daily $PM_{2.5}$ concentration. To achieve this, the author learned the relationship of the AOD image data to the ground-level measurements using an RF. Various data sources described in Section 2.1 were used by T. Li et al., 2017 to train a DBN to predict the grid cell pollutant concentration. The daily average temporal and spatial $PM_{2.5}$ ground-level concentration across the USA was predicted by Di et al., 2019. The researchers used an RF, a GBR, and an ANN as an ensemble. Each model separately predicted the target grid cell and incorporated the neighboring cells' pollutant concentration. The individual prediction was then combined with a generalized additive model that weighted each prediction specific to the location (e.g., urban area) and time (e.g., season). Additionally, the authors analyzed the relative predictor performance for the different base learners.

To summarize this section, it can be concluded that while linear models and support vector regressors (SVRs) were used to predict the air pollutant concentration of a single station, for multiple stations, the researchers often relied on more complex recurrent neural networks architecture (with the majority of the revised articles using the LSTM) or 1D-CNN. In addition, mapping the AOD data to the ground-level grid cells was often preformed using the RF.

2.5. Model validation and results

The current section describes how different authors evaluated their models and the results they achieved. To ensure credible scientific results, it is common in ML to develop the learning algorithms on a training set and to evaluate them on a separate test set that the algorithm did not "see" before. Many authors split their data set by year, e.g., the training was performed on previous years to predict the following year (e.g., Biancofiore et al., 2017; Zhou et al., 2019), the data was randomly split and cross-validated (e.g., Castelli et al., 2020; Kleine Deters et al., 2017; Zhang et al., 2021) or cross-validated over the ground-measurement stations (Blond et al., 2003; Qiao et al., 2019). The validation setup and results of the studies that predicted future pollutants are presented in Table 2.3. While many authors compared various settings (e.g., different time steps for look back and horizon, different error measures), only results comparable to other studies are shown.

As seen in Table 2.3, most studies predicted the next hour's pollutant concentration and evaluated their performance with either the root mean squared error (RMSE) or the mean absolute error (MAE). In addition, many researchers calculated different metrics to compare their results (Saez and Barceló, 2022 achieved a mean absolute percentage error of 38.53% for the exact prediction). The look back of time steps varied from 0 (current) to 20 days and from 2 to 72 hours. The MAE value greatly varied from 2.94 $\mu g/m^3$ to 14.63 $\mu g/m^3$ for the one-hour ahead single station forecast and from 14.08 $\mu g/m^3$ to 23.97 $\mu g/m^3$ for the next 6 hours. Most studies employed a variation of the LSTM model. Even though Table 2.3 shows the results for predicting $PM_{2.5}$, many authors predicted different pollutant species like CO, sulfur dioxide (SO_2) (e.g., Saez and Barceló, 2022). Predicting horizon intervals like the average value of the lead hours 7-12 is also a common technique to evaluate the performance (X. Li et al., 2017). On the other hand, the number of time steps for the look back was sometimes investigated via auto-correlation (Tao et al., 2019). It should be noted that most of the studies shown in Table 2.3 used the historical pollutant concentration of the target site, and only few did not include these values as inputs (Kleine Deters et al., 2017; Saez and Barceló, 2022). The model employed by Liang et al., 2018 additionally allows the visualization of the learned spatial relationship between the different

Table 2.3: The table gives an overview of the different evaluation setups of the different studies. The main algorithm is presented with the number of *output* stations, the unit of each time step hour (h) or day (d), the past time steps that were considered (look back), and the future time step of $PM_{2.5}$-concentration in $\mu g/m^3$, that are predicted (horizon). A $\emptyset$–metric shows that the studies averaged their result across all target stations

Article	Algorithm	Num. stations	Unit	Look back	Horizon	Metric	Error value
Zheng et al., 2015	Ensemble	22	h	3	1-6	MAE	23.7
Biancofiore et al., 2017	Elman Network	1	d	1	1-3	r	0.89
Kleine Deters et al., 2017	CGM	2	d	0	0	$\emptyset$–MAE	15.3
X. Li et al., 2017	LSTM+FC	12	h	8	1	MAE	5.46
Liang et al., 2018	Attention network	1	h	6	1-6	MAE	14.08
Huang and Kuo, 2018	CNN-LSTM	1	h	24	1	MAE	14.63
Zhou et al., 2019	SVR	5	h	4	1	RMSE	4.49
Zhao et al., 2019	LSTM+FC	1	h	6	1-6	MAE	23.97
Du et al., 2019	CNN-GRU	36	h	9	1	MAE	9.96
Tao et al., 2019	CNN-GRU	1	h	8	1-2	MAE	10.48
Qin et al., 2019	CNN-LSTM	1	h	72	1-24	RMSE	14.30
Qiao et al., 2019	SAE-LSTM	1	d	20	1	MAE	3.88
Castelli et al., 2020	SVR	1	h	2	1	R^2	0.64
Chang et al., 2020	LSTM	1	h	72	1	MAE	2.94
Zhang et al., 2021	Bi-LSTM	13	h	8	1	$\emptyset$–MAE	4.92
Mao et al., 2021	TS-LSTM	12	h	12	1-12	$\emptyset$–RMSE	20.00
Zeng et al., 2022	LSTM	1	h	N/A	1	MAE	3.45
Akbal and Ünlü, 2022	LSTM	5	d	3	1	$\emptyset$–MAE	5.56
Tian et al., 2022	DBN	1	h	24	1	RMSE	14.06
Jin et al., 2022	VB-GRU	1	h	24	1-24	MAE	19.78
Saez and Barceló, 2022	GLMM	$\sim$30	d	0	0	RMSE	5.48

Table 2.4: The table presents the results of the different studies that estimate the daily average ground level $PM_{2.5}$ concentration in an area. The spatial resolution corresponds to the approximate size of one grid cell.

Article	Algorithm	Spatial resolution	RMSE
Hu et al., 2017	RF	12×12 km^2	1.78
X. Meng et al., 2021	RF	1×1 km^2	16.3
T. Li et al., 2017	DBN	10×10 km^2	13.03
Di et al., 2019	Ensemble	1×1 km^2	2.78

stations. Whereas the results in Table 2.3 mainly addressed the problem statement of future air pollutant prediction, Table 2.4 presents the achievements of estimating daily average ground level pollutant concentration from satellite images over an area. Considering the more difficult task of predicting the average concentration of a grid cell with higher spatial resolution, Di et al., 2019 outperformed the other authors with an RMSE of 2.78 μg/m^3. While the results of all presented studies are given in μg/ m^3, Muthukumar et al., 2021 estimated the daily average $PM_{2.5}$ concentration of ground-level measurement stations of the following two days. The RMSE for this study was 0.000751 parts per billion and, therefore, not directly comparable to the other studies.

In this section, the validation and results of various researchers were presented. As the predicted horizon grows or the resolution of a spatial grid is increased, the challenge for an accurate prediction increases. Furthermore, it is clear that even though comparable articles were presented, the diversity of different experimental setups made comparing the studies difficult. Nevertheless, the presented studies offered a comprehensive overview of applied algorithms and their success.

2.6. Synopsis

The discussed sections thoroughly explore air pollution studies, delving into data considerations, preprocessing techniques, predictive algorithms, and model evaluations. Section 2.1 shows how researchers utilize ground-level pollutant measurement stations, recording harmful gas concentrations in μg/m^3 or ppb. Meteorological factors and additional features, including time-related variables and data from neighboring stations, often shape the different datasets. The data sets include diverse measurement times, ground-level stations, and sampling frequencies. Satellite images, particularly AOD measurements, contribute to estimating ground pollutant concentrations. Next, Section 2.2 emphasizes the significance of preprocessing for effective ML use in air pollutant prediction. Challenges like categorical variables and numerical scale consistency are addressed. At the same time, methods for handling missing data and transforming time-related features are discussed, highlighting the crucial role of preprocessing in algorithm performance enhancement. The subsequent part of the Chaper 3 delves into predictive algorithms, showcasing a spectrum of models from single-station predictions to spatial grid modeling. Various studies implement ML algorithms like linear regression, SVM, and ensembles, with a notable emphasis on LSTM networks. Spatial and temporal relationships are explored through ensembles, decision tree regressors, and complex neural networks, demonstrating the evolving landscape of predictive modeling techniques in air pollutant research. Next, Section 2.5 outlines standard ML practices, emphasizing reliable model assessment through separate training and testing sets. Notable variations in validation setups, prediction horizons, and evaluation metrics, such as MAE and RMSE, are highlighted. The transition from predicting future air pollutant concentrations to estimating daily average ground-level concentrations from satellite images is discussed, showcasing the challenges of increasing spatial resolution.

These sections collectively present a holistic view of air pollution studies, covering data considerations, preprocessing steps, predictive algorithms, and model evaluations. Building upon the insights garnered from this section, the following chapter applies the gained knowledge, outlining the specific approaches taken to preprocess data, implement predictive algorithms, and evaluate models in the context of air pollution research.

3 Chapter Materials and Methods

In the following chapter, Section 3.1 outlines the technological framework and software tools utilized for model development, setting the stage for the subsequent methodological components. Afterward, Section 3.2 focuses on the acquisition, description, and preprocessing of different data and gives an overview of the inputs fed to the different machine learning (ML) models. In Section 3.3, a detailed framework for model development, refinement, and assessment is outlined, creating a robust experimental foundation for the study of air pollutant prediction.

3.1. Development environment

The underlying Latex template used to format this thesis is based on Navarro-Guerrero, 2014. Depending on the computing workload, the different programs, and scripts are either run locally on a Laptop with 8 GB RAM and 4 CPUs utilizing 2.5 GHz each or on a high-performance cluster electively with a GPU or CPU node. For the development of the different scripts and programs, Ubuntu 22.04 is used as the operating system. Considering its wide range of application possibilities and personal experience, Python 3.10 is used to implement the different programs and scripts. Apart from the provided Python standard libraries, various other libraries and frameworks are incorporated. They include netCDF4 (Whitaker, 2023) and xarray (xarray developers, 2023)) to handle NetCDF files used for grids of satellite data or grid forecasts and numpy (Harris et al., 2020), pandas (McKinney, 2010) and scikit-learn (Pedregosa et al., 2011) to further preprocess and prepare the collected data. Furthermore, scikit-learn provides a variety of different ML algorithms that are used in this thesis and tensorflow 2.11 (Martín Abadi et al., 2015) is utilized as a framework to implement artificial neural networks (ANNs). When missing values are expected in the inputs, the XGBoost library (Chen and Guestrin, 2016) is used in favor of scikit-learn since the implementation is able to handle missing values naturally. The optimization of the hyper parameters (HPs) belonging to the different learning algorithms is performed using SMAC3 (Lindauer et al., 2022). The different aspects of the results are visualized using Matplotlib.

3.2. Data

While Section 3.2.1 outlines the various steps taken to acquire, describe and harmonize suitable data to answer the overall research question, Section 3.2.2 details the steps taken to refine and prepare the raw data, addressing challenges such as missing values, categorical variables, and numerical scaling to ensure optimal compatibility with ML algorithms.

3.2.1. Acquisition and description

For this research, data from the following three different primary sources were merged: Deutscher Wetterdienst (DWD) for meteorological data, Umweltbundesamt (UBA) for the different air pollutants, and Copernicus Atmospheric Monitoring Service (CAMS) for the air pollutant forecasts. The in-situ observations from DWD, the metadata for the stations of UBA and the CAMS regional forecast were accessed via web interface on Deutscher Wetterdienst, 2023, Umweltbundesamt, 2023b and Copernicus, 2023, respectively, the in-situ observations for UBA were obtained on demand through e-mail correspondence. The latter can be further subdivided into measurements from federal (Umweltbundesamt, 2023a) and state governments (Bundesländer, 2023). To accelerate the process of downloading each DWD station manually, a simple web scraper was implemented to automate the process. Table 3.1 gives insights into the distribution of the different features collected from UBA and DWD.

All data was available at an hourly time resolution, spanning from 2020-03-01 to 2022-12-31. For the CAMS data, a spatial grid spanning over Germany from 55.292° north, 5.669° west, 47.339° south, and 15.249° east with a spatial resolution of 0.1° was acquired. The closest DWD station was fused with the ground-level pollutant station to add the meteorological factors. Additionally, the CAMS forecast in which spatial grid cell the UBA station is located was added and aligned over the time dimension. Since this research evaluates the improvement of regional forecasts in the local context of an urban environment, eleven major cities were chosen as data subsets. Each city is considered a separate study site, and each study site includes pollutant measurement stations within a radius of 50 kilometers of the city center as in Akbal and Ünlü, 2022. Note that an individual station can be used in multiple study sites. The number of different stations for each pollutant varies from city to city and can be found in Table 3.2. fine particulate matter with a diameter $< 2.5\mu g/m^3$ ($PM_{2.5}$) was chosen as the target pollutant because of its most severe effect on human health (European Environment Agency, 2023). A second pollutant was selected due to the difficulty in predicting its future concentration, which is additionally displayed in Table 3.2. The relative error per study site is expressed by

Table 3.1: Statistics for the different measurements from all available ground-level stations are displayed. The pollutants were measured by the UBA, and the meteorological measurements were taken by the DWD. The statistics of the various variables used as inputs differ notably. In particular, the pollutant measurements' minimal (Min) and maximal (Max) values stand out and can be denoted as measurement errors.

Variable	Min	Max	Mean	Median	Mode	Std	Variance
NO	-6.53	624.12	10.97	2.14	0.62	22.07	487.24
$NO2$	-6.66	345.26	21.13	16.95	2.00	16.06	257.86
$PM_{2.5}$	-9.94	2445.58	10.10	8.13	5.00	7.54	56.91
PM_{10}	-6.90	1083.50	16.19	13.60	1.50	12.94	167.46
O_3	-3.94	222.69	49.05	48.08	0.60	30.36	921.79
SO_2	-9.66	675.32	2.14	1.15	0.80	4.76	22.67
Precipitation [mm]	0.00	51.80	0.07	0.00	0.00	0.48	0.23
Temperature [C]	-24.00	39.20	11.33	11.10	9.00	7.77	60.43
Relative humidity [%]	11.00	100.00	73.67	78.00	94.00	19.02	361.88
Wind direction [degree]	0.00	360.00	190.60	210.00	220	-	-
Wind speed [m/s]	0.00	20.20	3.21	2.90	2.10	1.89	3.59

calculating the standard score for the mean absolute error (MAE) between CAMS prediction and ground-level measurements.

Table 3.2 compares the relative MAE of the CAMS prediction in the different cities chosen as study sites. It can be seen that the prediction of $PM_{2.5}$ concentration yields the lowest and O_3 the highest relative error over all cities. However, the standard score for O_3 is highly influenced by a single station in Stuttgart and does not reflect the performance of the CAMS prediction achieved in all other cities. Excluding Stuttgart from the data set results in the lowest relative error after particulate matters (PMs) over all cities. Finally, NO_2 was chosen over SO_2 as the second target pollutant because of the extensive measurement in all evaluated cities. The following graphics show the average distribution of the pollutant concentration for the target pollutants $PM_{2.5}$ and NO_2 concerning the month of the year, day of the week, and hour of the day. The data basis for the different figures is composed of the first two years of the data described in Table 3.2.

Figure 3.2 clearly shows a similar pattern in average pollutant concentration for the target pollutants. Both have their average peak concentration during March and the average lowest during July. Even though similar patterns can be seen, the $PM_{2.5}$ concentration seems to be more influenced (+26.82% during March and -33.76% during July) compared to the NO_2 concentration (+15.23% in March and -25.90% in July). In contrast, the NO_2 concentration increases more rapidly over the mean represented as dashed horizontal line (between August and September)

Table 3.2: Relative mean absolute error of the CAMS forecast for different pollutants and cities. N denotes the number of stations per calculated error. Even though ozone (O_3) has the highest relative error over all cities, it can be seen that a single measurement station highly influences this error in Stuttgart. Furthermore, the number of measurement stations per pollutant varies from 32 for sulfur dioxide (SO_2) to 191 for nitrogen dioxide (NO_2) overall evaluated cities.

	$PM_{2.5}$		PM_{10}		NO		NO_2		O_3		SO_2	
City	MAE	N	MAE	N	MAE	N	MAE	N	MAE	N	MAE	N
Stuttgart	0.50	15	0.52	13	0.69	15	0.94	15	15.70	1	-	0
Berlin	0.48	21	0.62	21	0.75	26	0.75	26	0.57	14	1.06	2
Dortmund	0.59	10	0.60	23	0.59	26	0.78	26	0.60	12	1.12	4
Duesseldorf	0.58	12	0.57	35	0.59	36	0.81	36	0.58	15	1.08	5
Frankfurt	0.58	13	0.62	17	0.71	21	0.85	21	0.65	13	2.76	9
Hamburg	0.49	6	0.59	12	0.66	14	0.74	16	0.52	8	0.58	6
Hannover	0.50	4	0.60	4	0.91	4	0.95	4	0.70	2	-	0
Koeln	0.57	7	0.56	21	0.58	21	0.81	21	0.59	11	-	0
Leipzig	0.47	4	0.59	10	0.66	11	0.76	11	0.63	8	0.72	4
Muenchen	0.64	6	0.57	5	0.61	6	0.86	6	0.66	5	-	0
Nuerenberg	0.55	5	0.60	8	0.58	9	0.71	9	0.60	5	1.30	2
	$\overline{x}$	$\sum$	$\overline{x}$	$\sum$	$\overline{x}$	$\sum$	$\overline{x}$	$\sum$	$\overline{x}$	$\sum$	$\overline{x}$	$\sum$
All	0.54	99	0.58	169	0.67	189	0.82	191	1.98	94	1.23	32

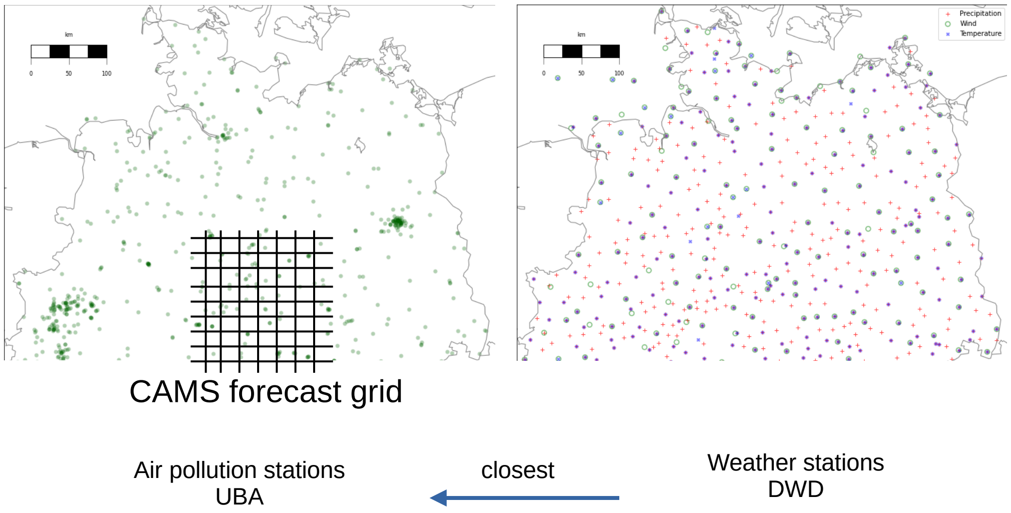


Figure 3.1: A schematic visualization of how the three different datasets are merged can be seen. The closest weather station for precipitation (red crosses), temperature (blue crosses), and wind (green circles without filling) on the right site is merged with the closest pollutant ground-level measurement station displayed on the left side as green circles. Additionally, the CAMS forecast grid cell, where the pollutant station is located, is added as input data per station.

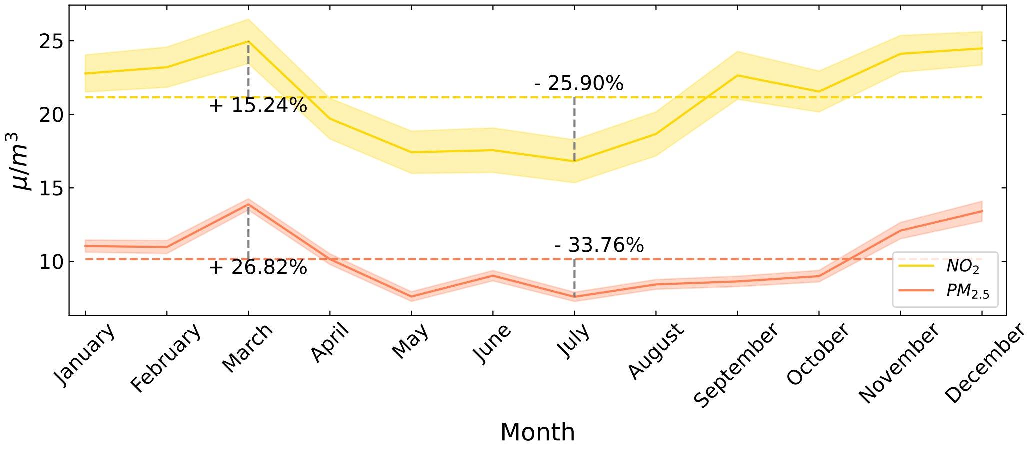


Figure 3.2: The monthly average pollution concentration and 95% confidence interval of $PM_{2.5}$ and NO_2. Both pollutants show a higher concentration level during the winter periods. The horizontal dashed line depicts the yearly average concentration.

than the $PM_{2.5}$ concentration (between October and November). Additionally, the 95% confidence interval shows a wider spread for NO_2 than for $PM_{2.5}$, indicating a higher challenge for predicting future concentrations.

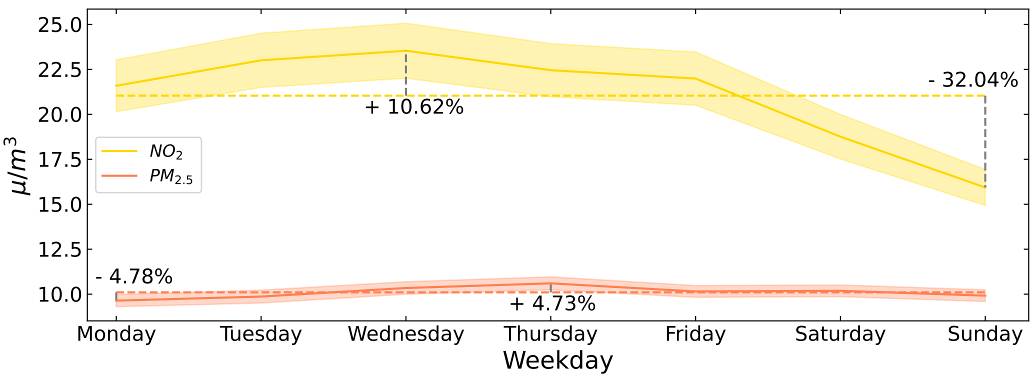


Figure 3.3: The average pollution concentration and 95% confidence interval of $PM_{2.5}$ and NO_2 per weekday. While the mean concentration of $PM_{2.5}$ only shows a marginally changed development over the week, NO_2 shows a trend of reduced pollutant concentration towards the weekend. The horizontal dashed line depicts the weekly average concentration.

Figure 3.3 shows the mean pollutant concentration per day of the week. The NO_2 concentration depicts a clear pattern with a peak concentration (10.62% above average) on Wednesdays in the middle of the usual 5-day working week and weekly low on Sundays (32.04% below average). The amount of combustion engine traffic might explain the concentration variation. On the contrary, the $PM_{2.5}$ concentration does not show similarly distinctive patterns, with the minimum and maximum concentration per hour only varying about 4.7% below and above average during Mondays and Thursdays, respectively. Again, NO_2 shows a wider spread than $PM_{2.5}$ for every weekday.

The hourly mean concentration of $PM_{2.5}$ and NO_2 can be seen in Figure 3.4. While NO_2 shows a clear pattern regarding the hour of the day, this pattern is less distinctive for $PM_{2.5}$. The lowest mean NO_2 concentration can be observed at 4 o'clock (28.09% below average), and the mean peak concentration can be observed 4 hours later at 8 o'clock(18.91% above average). An additional peak can be noticed between 18 and 24 o'clock in the evening. Similarly to NO_2, the $PM_{2.5}$ shows a peak concentration in the morning hours at 9 o'clock (8.42% above average) but opposed to NO_2 has an observed negative peak concentration at 17 o'clock. The 95% confidence interval for NO_2 depicts, similar to Figure 3.2 and Figure 3.3, a wider spread in contrast to $PM_{2.5}$. Furthermore, the spread of the NO_2 concentration is increasing towards the second half of the day. The figures above show how periods and an adapting environment can affect the pollutant concentration of $PM_{2.5}$ and NO_2, with the latter being more influenced by external

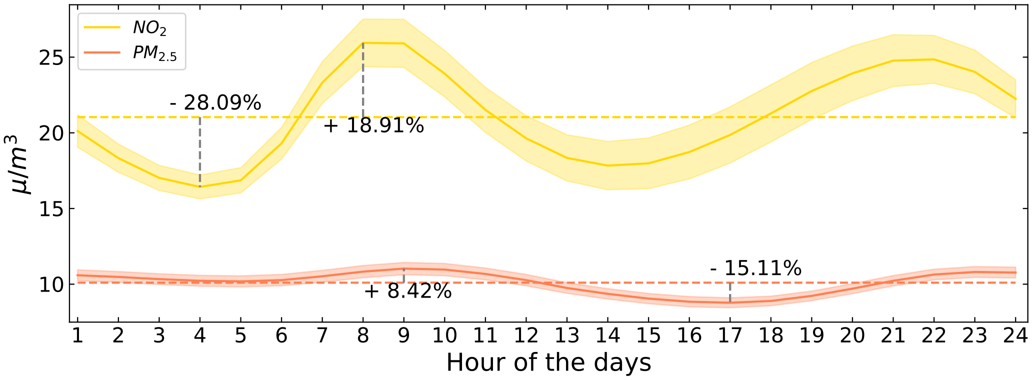
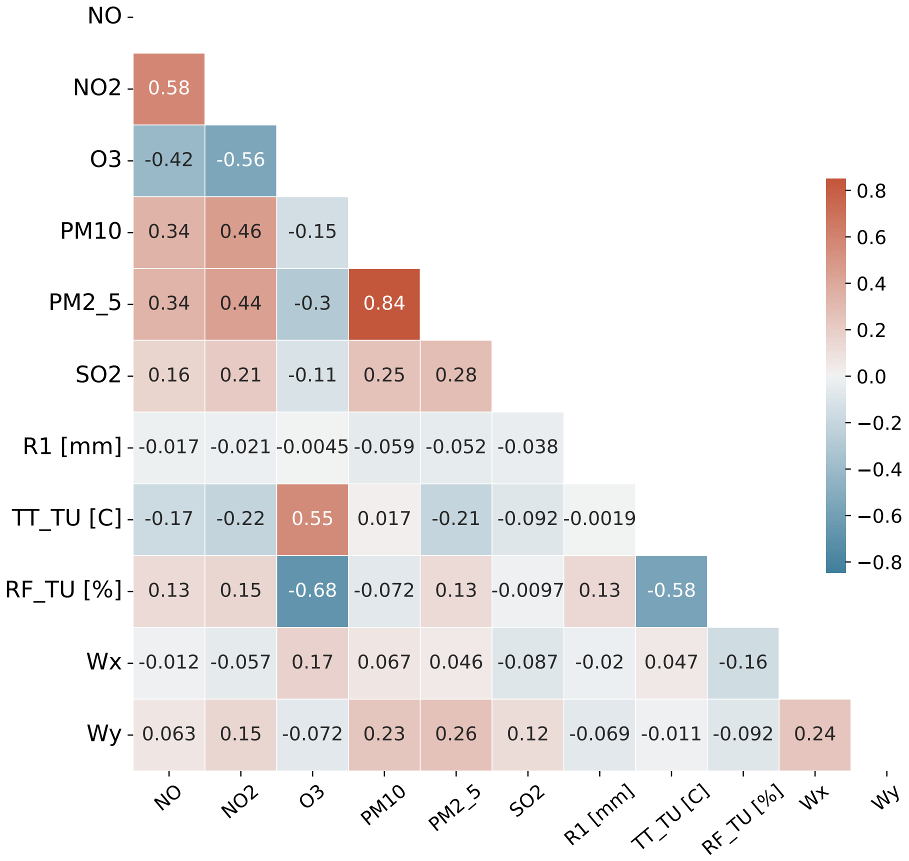


Figure 3.4: The average pollution concentration and 95% confidence interval of $PM_{2.5}$ and NO_2 for each hour of the day. Again, even though similar patterns can be seen in the average concentration distribution per hour of the day, the relative average change of NO_2 concentration during the day is higher than for $PM_{2.5}$. The horizontal dashed line depicts the daily average concentration.

factors. While the day of the week and the hour of the day are the most influencing factors for NO_2, $PM_{2.5}$ is influenced most by the month of the year. While the high variation of NO_2 can most probably be explained by human behavior (e.g., cars with combustion engines), $PM_{2.5}$ might be most affected by meteorological factors.

Figure 3.5 depicts a correlation matrix of the measured inputs displayed as a heat map. The highest (positive) linear correlation exists between $PM_{2.5}$ and PM_{10}. The target pollutant $PM_{2.5}$ is additionally been correlated with the other pollutants NO, NO_2, O_3 and SO_2. It also shows less distinct correlations with the meteorological measurements, temperature (TT_TU[C]), relative humidity (RF_TU[%]), and the one wind vector (Wy). The target pollutant NO_2 has the highest linear correlation with NO with a small margin to O_3. An intermediate correlation with the other pollutants can also be observed. Additionally, it slightly correlates with every meteorological factor except precipitation (R1 [mm]) and one wind vector (Wx).



	NO	NO2	O3	PM10	PM2_5	SO2	R1 [mm]	TT_TU [C]	RF_TU [%]	Wx	Wy
NO											
NO2	0.58										
O3	-0.42	-0.56									
PM10	0.34	0.46	-0.15								
PM2_5	0.34	0.44	-0.3	0.84							
SO2	0.16	0.21	-0.11	0.25	0.28						
R1 [mm]	-0.017	-0.021	-0.0045	-0.059	-0.052	-0.038					
TT_TU [C]	-0.17	-0.22	0.55	0.017	-0.21	-0.092	-0.0019				
RF_TU [%]	0.13	0.15	-0.68	-0.072	0.13	-0.0097	0.13	-0.58			
Wx	-0.012	-0.057	0.17	0.067	0.046	-0.087	-0.02	0.047	-0.16		
Wy	0.063	0.15	-0.072	0.23	0.26	0.12	-0.069	-0.011	-0.092	0.24	

Figure 3.5: A correlation matrix of the measured inputs is displayed as a heat map. The target pollutants $PM_{2.5}$ and NO_2 highly correlate with PM_{10} and NO, respectively. Additionally, both correlate positively or negatively with the other pollutants, with an absolute correlation between 0.21 and 0.56. Furthermore, the target pollutants correlate slightly with the meteorological factors except precipitation (R1 [mm]), including one wind vector (Wy). While the $PM_{2.5}$ has the strongest correlation with the wind vector (Wy), NO_2 correlates most negatively with the temperature (TT_TU[C]).

3.2.2. Preprocessing

This section describes the various preprocessing steps taken to transform the raw data to be used. One metadata file was created to represent the information for UBA and DWD. For this, the UBA specific metadata was downloaded from a separate source (Umweltbundesamt, 2023b), and stored in a file which includes for every station the station code, geographical location, measurement start, and end date, the station classification (background, industrial, traffic) and the area classification (rural, suburban, urban), and the information about which pollutant was measured at the particular station. For the DWD stations, a metadata file was created with the information given in the data source (Deutscher Wetterdienst, 2023). Besides the station code, the file includes information on the location, the start and end measurement date, and the measurement objective of the specific station (wind, precipitation, or temperature). As mentioned in the previous chapter, the closest DWD stations for the meteorological factors were merged by distance to the respective UBA ground-level station. A combined metadata file was created to store the pollutant station-related information, the closest meteorological station, and the corresponding distance, which was used to merge the correct measurement data subsequently.

As the first step, the different pollutant measurements provided in individual files are merged per station. Then, the raw ground-level UBA measurements missing values or erroneous values (indicated with -999) were replaced with a “not a number” value. A date-time index was constructed for each hour from the given “Date” and “Time” columns. The “PM1” and “PM2” columns were renamed to “PM10” and “PM2_5”, respectively. The time range of the stations was narrowed between 2020-03-01 and 2022-12-31. If the target pollutant (e.g., $PM_{2.5}$) has more than 10% of missing or erroneous measurements at the target station to process, the station was not included in this study. Next, the closest DWD ground-level measurements were aligned, matching the same time as the UBA observations. Moreover, the CAMS prediction for the location of the ground-level UBA station was added and aligned over the time dimension. The meteorological missing or erroneous values (indicated with -999) were also replaced with NaN. Furthermore, white spaces in the column names were removed, and the corresponding unit per meteorological factor was added (e.g., “ F” was converted to “F [m/sec]”). Additionally, all columns that contain data irrelevant to the learning algorithms were dropped. Time-related features similar to Castelli et al., 2020 were added to the data set. For this, the weekday or month was mapped to an integer value (e.g., January:0, February: 1) and then transformed on the unit circle, e.g., into the components cos($2\pi\times$month/12) and sine($2\pi\times$month/12). This gives the subsequently utilized ML model the ability to learn the recurring patterns of the month, weekday, and hour of the day, which are evident in Figure 3.2, Figure 3.3 and Figure 3.4, respectively. In the next step, the data set was split into inputs (all meteorological factors, all pollutants, time features, CAMS prediction) and outputs (the target pollutant). The missing values in the input data were filled depending on the data type. For the missing meteorological factors (added from the closest ground-level measurement stations), the next station with a valid measurement was used if the station was at most 100 km from the target pollutant station. Missing or erroneous pollutant measurements were filled, similar to Hu et al., 2017, using the CAMS prediction of the particular pollutant for that time and location. The wind direction was represented in degree and might therefore be difficult to interpret by the learning algorithms, e.g., the difference of 1° from 359° to 0° is not reflected well in this representation. Therefore similar to Kleine Deters et al., 2017, the wind degree was first transformed from polar to Cartesian coordinates and additionally multiplied by the wind speed.

Next, the data set until 2022-05-31 was reserved for training; the remaining data was for testing purposes. Both data sets were rearranged into daily samples comprising the current hour and past 23 hours as inputs and the next 23 hours of the target pollutant as output. Similar to Kleine Deters et al., 2017, samples that still included missing values were dropped from the data set. As in Chang et al., 2020; Tao et al., 2019; Zhou et al., 2019, the training data was standardized by calculating the mean and the standard deviation of the training data to subtract the mean from training and test data and additionally divide both sets by the standard deviation of the training data. For the ANNs, two additional data sets were constructed by re-sampling the data to a 2-hour and 4-hour measurement frequency. For all other learning algorithms, the training and test data was flattened by merging the lookback and feature dimension so that (n_samples, n_lookback, n_inputs) becomes (n_samples, n_lookback×n_inputs).

3.3. Experimental Set-Up

In the following section, the machine learning (ML) algorithms utilized in this study are proposed, followed by a description of how the selection procedure was performed and how the proposed algorithms were optimized using hyper parameter optimization (HPO). Subsequently, the different Scenarios that serve as the basis for evaluating the different algorithms' performance are introduced. Moreover, a description of how the different predictions were combined is presented. The chapter is finalized by describing the evaluation scheme used to measure the performance of the different predictive models.

3.3.1. Proposed models

Since the goal of this research is to evaluate how regional forecasting models can be enhanced in local urban environments using ML, various learning algorithms were utilized so that the answer to the research question does not depend on the capability of a single ML model class. Orientating on the implemented ML algorithms revised during the literature review and summarized in Table 2.3, artificial neural networks (ANNs) composed of long-short term memory (LSTM) (Akbal and Ünlü, 2022; Chang et al., 2020; Huang and Kuo, 2018; X. Li et al., 2017; Mao et al., 2021; Qiao et al., 2019; Qin et al., 2019; Zeng et al., 2022; Zhang et al., 2021; Zhao et al., 2019) were utilized for this research. Moreover, convolutional neural network (CNN) (Du et al., 2019; Huang and Kuo, 2018; Qin et al., 2019; Tao et al., 2019) were the most often used algorithms and therefore also utilized in this work. Notably, the CNN was always used in combinations with other layer types by the researchers (see Table 2.3). In addition, ML algorithms that were successfully applied by Bertrand et al., 2023 using model output statistic (MOS), namely the Lasso, Ridge regression and gradient boosting regressor (GBR) were implemented. Furthermore, a support vector regressors (SVRs) with a linear kernel (as in Castelli et al., 2020; Zhou et al., 2019) was evaluated. Whenever missing values were expected in the input data, extreme gradient boostings (XGBs) (set up in a similar way to the GBR from literature) were used. A comprehensive explanation of the previously mentioned ML algorithms (excluding LSTMs, CNNs and XGB) can be found in Hastie et al., 2009. For explanations on LSTMs and CNNs, one can revise Goodfellow et al., 2016. A description of XGB and the underlying implementation can be found in Chen and Guestrin, 2016.

3.3.2. Feature selection

To reduce the data input dimension and simplify the prediction task for the ML algorithms, a subset of input features (predictors) most relevant to the model's response were chosen. This step not only reduces the computation time but also has the potential to increase the performance of the different learning algorithms. One way to reduce the dimension of the input data is to identify essential predictors from the correlation between input and target pollutants displayed in Figure 3.5 of Section 3.2.1. However, the correlation matrix only depicts linear dependencies between the different predictors and does not show non-linear dependencies. Feature importance methods that can show non-linear relationships include the permutation importance introduced by Breiman, 2001 and refined by Fisher et al., 2019, which shuffles the values of the different features to measure the impact on the model performance. Additionally Shapley values, which estimate how much each predictor value alone and in coalition with other feature values contribute to the response variable can be used to show non-linear dependencies Molnar, 2020. While these methods are promising, there is a lag in implementations capable of handling time series data.

Since the number of combinations for the best subset of n features is 2^n, a manual search over all combinations incorporating the different evaluated ML algorithms is impossible. Additionally, the hyper parameter (HP) configurations of the different ML algorithms influence how the different predictors are processed. Therefore, this research treats the search for the best subset of features as a search space problem with each feature included or not using the HPO methods introduced in the following Section 3.3.3.

3.3.3. Hyper parameter optimization

In the context of ML, a HP can be defined as a parameter that influences how the algorithms learn during training without being modified by the learning algorithm itself (Goodfellow et al., 2016). Due to the vast amount of adjustable HPs in a ANN, finding the optimal set and their corresponding values can be especially challenging. Domain knowledge of the underlying data can help to reduce the set of HPs and narrow their corresponding ranges. The capacity of a model enables or restricts its capability to match the complexity of a given task, resulting in underfitting when the capacity is too low and overfitting when the capacity is too high (Goodfellow et al., 2016). Adjusting the HPs is often a search for the suitable model capacity that matches a given task to minimize the generalization error. Standard techniques include the hand-tuning and automated HPO. The first requires expert knowledge of the effect of the different HPs on the specific task at hand and the interplay between them to be successful.

There are various methods for automatic HP search. A grid search can be regarded as the simplest one, in which one defines a finite set of values per HP, of which each combination value is evaluated on a validation set to find the best combination of HP in the configuration space. Since this strategy tests all combinations, grid search is only feasible for three or fewer tuning parameters. A good alternative to grid search is random search proposed by Bergstra and Bengio, 2012. In favor of defining a fixed set of values per HP as in grid search, value ranges, or sets are defined, from which specific values are sampled for each HP during the search, up to a predefined number of trials. This procedure has been shown to outperform the grid search on various tasks with less computational time (Bergstra and Bengio, 2012). They argue that since all combinations of HPs are tested in a grid search experiment, less important HPs allocate to many trials, and important ones need better coverage in their dimension. Taking this idea one step further, Bayesian optimization treats the search for the right HP configuration space as a ML problem itself (Russell, 2010). The task of the ML algorithm is to find the best HP configuration in a configuration space. It is seen as a trade-off between exploration (identifying HPs in uncertain areas), and exploitation (using HPs with which the model is confident) by Goodfellow et al., 2016.

One technique of Bayesian optimization (also known as HPO) can be found in sequential model-based algorithm configuration (SMAC) proposed by Hutter et al., 2011. SMAC overcomes the previous limitation of sequential model-based optimization procedures that were only capable of handling numerical HPs by applying an random forest (RF) as surrogate models. A concrete implementation of SMAC that is used in this research can be found in the software framework SMAC3 (Lindauer et al., 2022).

As Section 3.3.2 mentions, selecting beneficial input variables and the number of lookback timesteps are part of the HPO process. Since the number of HPs of the different ML algorithms mentioned in Section 3.3.1 greatly varies, two different strategies were employed to optimize each configuration space. In the first strategy, a maximum of 2000 trials with diverse HP combinations were sampled for the different ML algorithms (excluding the neural networks) using SMAC. A summary of the evaluated HPs can be found in Table 3.3.

Because the number of HPs in the configuration space for the neural networks is higher and the average training time per combination is longer, as for the other ML algorithms, a step-wise parameter search similar to Hinz et al., 2018 was applied. More specifically, the underlying data set was resampled from a 1-hour frequency to a lower resolution, corresponding to a 2-hour and 4-hour measurement frequency. Starting with the lowest resolution, 1000 parameter combinations were evaluated for every frequency resolution. After the first iteration of the 4-hour resolution data set, the value ranges of the different parameters were narrowed by choosing new lower and upper bounds for the next iteration. These bounds were identified from the 50 best-performing trials. For numerical values, these trials' mean and standard deviation were calculated per HP. The mean was the default value for the particular HP for the higher resolution; subtracting and adding the standard deviation from the mean was the lower and upper bound, respectively. The importance of categorical HP was evaluated using the number of occurrences of the different options in the 50 best-performing trials. If an option was represented less than 10% (5 times) in this trial, it was excluded in the following optimization iterations. All remaining options were weighted due to their relative number of occurrences in the first 50 trials. Moreover, binary variables and the number of LSTM layers were also treated as categorical HPs. This process was repeated to perform the last HPO on the original resolution. The following HPs were optimized during the search for the ANNs. Similar to the previously presented HPO process, data-specific parameters included the number of timesteps used as input and whether the different input features were included

Table 3.3: The value ranges or sets for the different HPs of the implemented machine learning algorithms are shown. All available HPs provided by the scikit-learn library (Pedregosa et al., 2011) were evaluated, if applicable. For value ranges, suggestions by the library above were considered. The library also provides a description of each of the HPs and how they are implemented.

Algorithm	Hyper parameter	Values
Lasso	alpha	[0, 15]
	tol	[0.00001, 0.001]
	precompute	{False, True}
	positive	{True, False}
	selection	{cyclic, random}
Ridge	solver	{svd, cholesky, lsqr, sag, lbfgs}
	tol	[0.00001, 0.001]
	alpha	[0, 15]
Linear SVR	loss	{epsilon_insensitive, squared_epsilon_insensitive}
	tol	[0.00001, 0.001]
	C	{0.001, 0.01, 0.1, 1.0, 10, 100, 1000}
GBR	loss	{squared_error, absolute_error, huber, quantile}
	learning_rate	[0.001, 1]
	n_estimators	[50, 500]
	criterion	{friedman_mse, squared_error}
	min_samples_split	[2, 10]
	min_samples_leaf	[1, 10]
	max_depth	[1, 10]
	max_features	{sqrt, log2, 1.0}
	n_iter_no_change	{10,100,1000,10000,100000}

in the learning process. Furthermore, the parameters batch size, initial learning rate, optimizer, loss, learning rate scheduler, and early stopping, which directly influence the learning process, were included in the search space.

Intuitively, the batch size controls, after how many samples a weight update is performed. The strength of the update is determined by the learning rate times the calculated loss between predicted and actual values, and is performed using the optimizer. The used scheduler reduced the learning by lr_new=lr_decrease*lr_old, where 0.7<lr_decrease<0.95. The configuration space included the starting epoch, the epoch frequency, and the learning rate decrease. Early stopping is a simple regularization parameter that terminates the training process after no improvement related to the validation loss is achieved for a definable number of epochs (patience, (Goodfellow et al., 2016)).

All ANN were trained for a maximum of 200 epochs. Since there is an arbitrary number of combinations for the neural network structures, types, and activation functions, the set and ranges of related HPs were roughly based on the results in the literature and summarized in Table 2.3. The following Table 3.4 outlines the initial configuration space of the HPs for the first iteration of the optimization process on the lowest data resolution (4-hour time step).

Table 3.4: The value ranges or sets for the different HPs corresponding to the ANNs are shown. The parameters were loosely based on the revised literature presented in Section 2.5 and the suggestion of the underlying deep learning framework. Note that the value ranges correspond to the first iteration of HPO calculated on the lowest data resolution.

Hyper parameter	Values	Dependent on
Batch size	[16, 64]	-
Optimizer	{Adam, Nadam, SGD, RMSProp}	-
Initial learning rate	$[5 * 10^{-4}, 1.5 * 10^{-3}]$	-
Loss	{huber loss, mean squared error}	-
Learning rate scheduler	{True, False}	-
Start epoch	[3, 25]	Learning rate scheduler == True
Learning rate decrease	[0.75, 0.95]	Learning rate scheduler == True
Every N epoch	[3, 10]	Learning rate scheduler == True
Early stopping	{True, False}	-
Patience	[3, 15]	Early stopping == True
CNN layer	{True, False}	-
CNN filter size	[2, 5]	CNN layer == True
Num. LSTM layers	[1,4]	-
Dense layer	{True, False}	-
Units/Filters	[16, 512]	Dense layer == True
Normalization layers	{True, False}	-
Dropout layers	{True, False}	-
Dropout	[0.0, 0.5]	Dropout layers == True

The HPs for the model architecture included parameters concerning the global structure of the model and layer-specific parameters. Parameters defining the global structure included whether or not a convolution was used as a feature extractor in the first layer (CNN layer). Furthermore, it was determined how many LSTM layers are used in the network (num. LSTM layers). Adding a fully connected layer between LSTM and the output layer was the final HP of the global model structure (Dense layer). The number of units (or filters for the CNN layer) was evaluated for every layer. Adding normalization after each layer was given as an additional option. Additionally, dropout was applied after each layer, varying from zero (no dropout) to 0.5. The dropout procedure is a regularization technique that randomly drops a fraction of units in the preceding layer by multiplying them with zero (Goodfellow et al., 2016). Finally, the filter size of the CNN layer was also part of the searching process with the filter size greater than one to a maximum number of past timesteps included in the training process.

Regardless of the learning algorithm, of all the available data shown in Table 3.1, only stations that measure the maximum number of input features described in Table 3.1 were considered. Each HP configuration was performed via cross-validation on a subset of three randomly chosen stations from all previously selected stations. The randomly selected stations were the same for all HP configurations. Notably, the HP search was only performed on data corresponding to the first 2/3 of the training time range. For validating the performance of the different ML algorithms, the last 1/3 of the training time range was used. In total, for the four different combinations corresponding to the target pollutants fine particulate matter with a diameter $< 2.5\mu g/m^3$ ($PM_{2.5}$) and nitrogen dioxide (NO_2), with or without Copernicus Atmospheric Monitoring Service (CAMS) as additional input, HP were optimized individually.

3.3.4. Scenarios

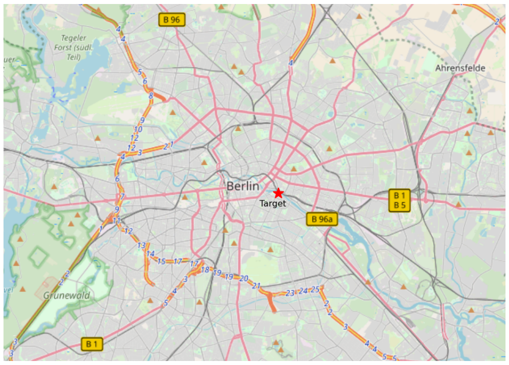


Figure 3.6: The map shows the experimental setup for Scenario 1 in the study area of "Berlin". The position of the target station is highlighted as a red star.

The section describes three scenarios to investigate the research question in detail. The data basis for all scenarios were the study sites across the eleven major cities, shown in Table 3.2. Each study site includes stations within a radius of 50 kilometers from the city center. Since particulate matter (PM) can travel great distances (Kim et al., 2015), this research chooses the same radius as in Akbal and Ünlü, 2022, to have a sufficient radius to incorporate all stations in the different cities and their surroundings. While Scenario 1 (S1) evaluated the performance of the different learning algorithms mentioned in Section 3.3.1 concerning a single station (for all stations inside the study area), Scenario 2 (S2) and Scenario 3 (S3) also incorporated neighboring stations. The latter excluded measurements at the target location, thus simulated the interpolation (or extrapolation) of spatial relationships or an average pollutant concentration inside the given study area. Furthermore, the use of MOS (including the CAMS forecast as input or not), was additionally examined for all three scenarios.

Scenario 1 incorporated up to 23 hours past pollutant concentrations measured at the particular station. Moreover, the current and past meteorological factors were included as input. For each station in every study area, a ML algorithm was trained to predict the next 23 hours of the targets pollutants concentration (either $PM_{2.5}$ or NO_2). The training was performed on the training data set (2020-03-01 to 2022-05-31). Subsequently, the learning algorithm predicted the validation data ranging from 2022-05-31 to 2022-12-31. Note that the data was already preprocessed for this scenario due to the various steps described in Section 3.2.2. The raw predictions were saved for later evaluations per study area and station. This process was repeated, applying MOS by additionally including the CAMS forecast of the particular grid cell, in which the target ground-level station was located (see Figure 3.1, left) as input.

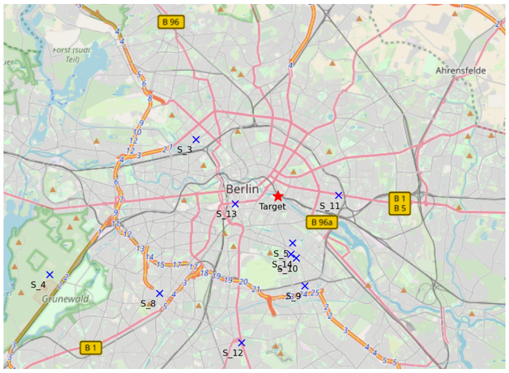


Figure 3.7: The map shows the experimental setup for Scenario 2 in the study area of "Berlin". The position of the target station is highlighted as a red star, and the neighboring stations are blue crosses. In this example, stations inside a radius of 25 kilometers are displayed. For the actual training, a 50-kilometre radius was applied.

Scenario 2 incorporated up to 23 hours of past pollutant concentrations measured at the neighboring stations to combine this information with the previously saved prediction performed at the target location. In accordance with the preprocessing steps described in Section 3.2.2, the prediction used as input was standardized using the z-normalization. Additionally, the meteorological factors closest to the target station were included as inputs. Similar to S1, a ML algorithm was trained for each station in every study area to predict the next 23 hours of the target pollutants concentration (either $PM_{2.5}$ or NO_2) at the chosen target station from every neighboring station. The subsequent training process was performed in the same way as in S1 (except for the additional inputs), and the resulting prediction was saved per neighboring station for later evaluation. Note that in the example shown in Figure 3.7, the target station is surrounded by neighboring stations in multiple directions, which could help to increase the performance. Other study areas or target stations might not have a similar beneficial station distribution.

Scenario 3 was evaluating the same setup as described in Figure 3.7 for S2, except that the historical pollutants of the target measurement station (or the predictions that incorporate these measurements) were excluded, so that the target point was interpolated over the spatial dimension. The target station measurement was only used to validate the final prediction error. The different ML algorithms were only trained on the neighboring stations, including the meteorological data from the target location. Each neighboring station might additionally include information on bearing and distance to the target station. Additionally, and in contrast to S1 and S2, a single ML model that incorporates all neighboring stations simultaneously was investigated. Here, the XGB model was used due to its native support to handle missing data. It is expected that using MOS by incorporating the CAMS forecast is particularly beneficial to improve the performance when the historical data at the target location is missing.

3.3.5. Combining the predictions

Even though the different approaches were already outlined in the previous section, they are clarified further in the following. Different approaches were applied to fuse the predictions of the neighboring stations included in S2 and S3. One major challenge during the fusion process was dealing with missing input data (e.g., one neighboring station has erroneous values during an evaluated period) since most implemented ML algorithms can not handle varying input dimensions or missing data natively. Therefore, the first approach to combine predictions was to train a single model for each neighboring station to predict the next 23 hours at the target location and average the predictions over all neighboring stations. A second approach to combine the various neighboring inputs was utilized using XGB since the underlying implementation can handle missing data natively. In this setup, one model was trained using all available inputs from the neighboring stations simultaneously and thus can learn the temporal and spatial relationships from the inputs. Both approaches only differed due to the inputs and training procedure described in the following section. While the first approach was applied to both scenarios, the second approach was only applied to S3.

3.3.6. Evaluation

The performance of each ML algorithm was evaluated at each station shown in Table 3.2 for $PM_{2.5}$ and NO_2 based on the previously saved predictions per station. As the primary performance metric, the mean absolute error (MAE), which is mainly used in literature for air pollutant regression tasks (see Table 2.3), was used. Moreover, in the case of predicting future pollutant concentration, the MAE was more accessible to interpret and gives a natural insight into the error of the evaluated model. Additionally, the root mean squared error (RMSE) and the coefficient of determination R^2 were assessed. Compared to the RMSE and the R^2, the MAE is less sensitive to outliers, making it more suitable for situations where outliers are less critical. The R^2 error can give additional information about the goodness of a fit, with a value of 1 indicating that all and a value of 0 indicating that the regressor has explained none of the variability in the data. In exceptional cases, when the predictions are worse than always predicting the mean of the observed data, the R^2 can be below 0. The R^2 is averaged over each of the 23 predicted hours. Supplementing measures to access the performance included various graphics and the mean signed deviation (MSD), which can show the bias of the models. As mentioned earlier, the overall performance of the different ML algorithms were compared against each other and whether MOS was applied or not. The CAMS prediction was the baseline for all scenarios. When local data was available (as for S1 and S2), the CAMS prediction was bias-corrected, similar to the difference method by EEA, 2023. More concretely, the average prediction error of the past four days between the CAMS prediction and the local measurements was either added or subtracted for future predictions. A comparison between the two approaches that combine predictions explained in the previous Section 3.3.5 was additionally applied. Finally, the achieved results were opposed to comparable results from the literature.

3.4. Synopsis

After introducing the development environment and accompanying software, the three underlying data sources Deutscher Wetterdienst (DWD) for meteorological information, Umweltbundesamt (UBA) for air pollutant data, and CAMS for air pollutant forecasts were presented. The data spans from March 1, 2020, to December 31, 2022, with hourly time resolution and includes a spatial grid over Germany. Eleven major cities were chosen as study sites, each with pollutant measurement stations within a 50-kilometer radius of the city center. The evaluated target pollutants were $PM_{2.5}$ and NO_2, with a detailed analysis of their average distribution across the month, weekday, and hour of the day. The preprocessing steps involved creating metadata files, merging pollutant measurements, handling missing values, aligning meteorological factors, and adding time features. Additional transformations were applied for the representation of wind direction and time features. The final dataset was divided for training and testing, standardized, and formatted to reflect daily samples for different time resolutions (1 hour, 2 hours, 4 hours). After the dataset was prepared, different ML algorithms, including Lasso, Ridge regression, GBR, SVR and ANN's with LSTM, and CNN layers were employed. To optimize the different ML algorithms HPO, challenges were addressed through various strategies, including SMAC, resampling data at different resolutions, and sequentially narrowing HP ranges. Three scenarios were explored, assessing the performance of ML algorithms for pollutant concentration prediction, with two fusion approaches proposed for the neighboring stations in S2 and S3. The evaluation primarily employed the MAE as a metric to measure the overall performance, supplemented by graphics and MSD. Furthermore, the impact of MOS was analyzed, and the results were benchmarked against the CAMS predictions and literature. In this context, the following section presents and arranges the results for the HP search, each scenario, and each target pollution.

4 Chapter Results

This section provides the results to answer the research question of how an established regional forecasting system can be improved in urban environments. As mentioned in Section 3.2.1 and Section 3.3.3, the results for the hyper parameter optimization (HPO) are based on a subset of three randomly selected stations from the study sites presented in Table 3.2. The data basis for all other results is the eleven study sites presented in the same table. The training and validation period for HPO was only performed on the training period from 2020-03-01 to 2022-05-31 (approximately 4/5). The period from 2022-06-01 to 2022-12-31 was reserved as test data for the three scenarios described in Section 3.3.4. The subsequent sections are ordered in the way the experiments were performed. First, the results of the found hyper parameters (HPs) of the different machine learning (ML) algorithms are presented. Second, the performance of the different algorithms is compared against the baselines (Copernicus Atmospheric Monitoring Service (CAMS)) for Scenario 1 (S1). Here, a more detailed view is made concerning each predicted pollutant, individual regressor, and hour of the day. Moreover, the best-performing learning algorithm is compared in a classification setup corresponding to the different pollutant indices shown in Table 1.1. Third, the results for Scenario 2 (S2) and Scenario 3 (S3) are presented and compared against the baselines. For S3, the specific cases of using a distance vector as supplementing input and incorporating all neighboring stations simultaneously in one model are additionally presented. This section closes by aggregating the results and raising questions for the next section.

4.1. Hyper parameter optimization

The following presents the found HP of the different learning algorithms. The HPO was performed in a single station setup similar to S1. The HPs, that are based on the data set that includes the CAMS prediction as additional input are presented below, the HPs configurations without model output statistic (MOS) can

Table 4.1: The found input features corresponding to Figure 3.5 for the current and past time steps are shown. The different inputs are displayed for each pollutant and learning algorithm. The tick indicates the inclusion of the feature. While the number of lookbacks varies for both pollutants, most algorithms included six or fewer hours in their best setup. NO_2 is the only input not used by any algorithm, and not a single feature is used by all. There are also patterns regarding the particular pollutant, e.g., the corresponding target pollutant (NO_2 or $PM_{2.5}$) is always included as input.

Input	$PM_{2.5}$					NO_2				
	Lasso	Ridge	SVR	GBR	ANN	Lasso	Ridge	SVR	GBR	ANN
Look back	5	5	6	2	19	20	9	1	2	6
NO_2						✓	✓	✓	✓	✓
NO										
O_3			✓			✓	✓	✓	✓	✓
$PM_{2.5}$	✓	✓	✓	✓	✓	✓				
PM_{10}	✓	✓		✓						
SO_2					✓					
Precipitation [mm]	✓	✓	✓	✓				✓		
Temperature [C]	✓	✓	✓	✓		✓	✓	✓		✓
Relative humidity [%]	✓	✓	✓	✓	✓			✓	✓	✓
Wx	✓	✓	✓	✓		✓	✓	✓	✓	✓
Wy	✓	✓	✓	✓	✓	✓		✓	✓	
sine [m]								✓	✓	✓
cosine [m]	✓		✓	✓		✓		✓	✓	
sine [wd]	✓	✓	✓	✓			✓	✓	✓	✓
cosine [wd]	✓	✓	✓			✓		✓	✓	✓

be found in the Attachment (Table A.1, Table A.2 and Figure A.1). The displayed HP shows the single best HP configuration for each pollutant and ML algorithm.

Table 4.1 presents the selection of different input features. The lookback time steps greatly vary, from one to 20, which nearly corresponds to the maximum number of 23 hours. Mostly, only the past six or fewer time steps were considered as input. While nitrogen monoxide (NO) was not used by any algorithm as input, sulfur dioxide (SO_2) was only used to predict fine particulate matter with a diameter $< 2.5\mu g/m^3$ ($PM_{2.5}$) utilizing the artificial neural network (ANN). All algorithms included the respective historical target pollutant, nitrogen dioxide (NO_2) was omitted in all ML algorithms predicting $PM_{2.5}$, and $PM_{2.5}$ was only incorporated by the Lasso algorithm to predict NO_2. While none of the algorithms predicting NO_2 relied on particulate matter with a diameter $< 10\mu g/m^3$ (PM_{10}), the Lasso, Ridge, and gradient boosting regressor (GBR) included the pollutant. The overall usage of the meteorological factors as input was more often relevant for predicting $PM_{2.5}$. Nevertheless, the wind vector (Wx) and the temperature were relevant in predicting NO_2. Precipitation, on the other hand, was only incorporated by the support vector regressor (SVR). For the time-related feature, the patterns could be more precise, even though most ML models include either the sine or the cosine of the features for the month ([m]) or weekday ([wd]). Table 4.2 shows the determined parameters of each ML algorithm during the HPO.

The initial search space of each result presented in Table 4.2 was defined in Section 3.3.3 and can be found in Table 3.4 for the ANN and in Table 3.3 for all other algorithms. Up to 3000 HP combinations were tested for the ANN and 2000 for the other ML algorithms. A more detailed description of the experimental setup can be found in Section 3.3.3. For the different HP, similarities and differences can be seen between the two predicted pollutants for each presented ML algorithm. For example, the Lasso algorithm was always precomputed using the Gram matrix, and the coefficients were not forced to be positive. For the Ridge regression, the regularization coefficient alpha was close for both pollutants (14.807459 and 14.990312, which lays close to the maximum of 15 as defined in Table 3.3). The L1 (epsilon insensitive) loss was used for both pollutants by the Linear SVR. Furthermore, the complexity of the individual trees of the GBR ensemble controlled by the minimum of samples per split and leaf and the maximum depth of the trees show similar results. In the case of the ANN, a outstanding difference concerning the initial search space can only be seen in the choice of optimizer. It is noteworthy that for the computation of the loss, the Linear SVR, GBR, and ANN all rely on calculations that are less sensitive to outliers. Figure 4.1 depicts the found architecture of the ANN corresponding to the target pollutants $PM_{2.5}$ and NO_2 when CAMS was included as input. The displayed HPs together with the HP shown in Table 4.2 complete the configuration space defined in Table 3.4.

Compared to the initial search space shown in Table 3.4, both networks are kept relatively simple concerning the number of layers and units per layer. Apart from that, the upper part of the network (before CAMS input is concatenated) of the network displayed on the right site used an additional convolutional layer as a feature extractor. Also, the regularization was partly achieved differently. For $PM_{2.5}$, layer normalization was incorporated in addition to dropout. On the other hand, the dropout value in the upper part of the network for predicting NO_2 is more than 15 times as high as for $PM_{2.5}$. The structure of the lower part of the networks is similar compared to each other. Both include an additional fully connected layer (FCL) with a similar number of units after the concatenating with the CAMS prediction. Table 4.3 shows the statistics describing the distribution of the mean absolute error (MAE) for the different ML algorithms. Table 4.3 is based on 2000 HP combinations for each of the presented algorithms. Similar patterns can be seen for both pollutants. For example, the Ridge regression shows the lowest mean and maximum MAE and the lowest standard deviation in both cases. Moreover, the Linear SVR shows the highest maximum value and standard

Table 4.2: The found HPs for the different ML algorithms are shown. The resulting HPs regarding the ANN architecture are separately displayed in Figure 4.1. For each algorithm, some similarities and differences can be found. Remarkably, the Linear SVR, GBR, and ANN all chose loss functions that are less sensitive to outliers.

Algorithm	Hyper parameter	Found value	
		$PM_{2.5}$	NO_2
Lasso	alpha	0.036334	0.194414
	tol	0.000872	0.000025
	precompute	True	True
	positive	False	False
	selection	random	random
Ridge	solver	cholesky	lsqr
	alpha	14.807459	14.990312
	tol	0.000346	0.000359
Linear SVR	loss	epsilon_insensitive	epsilon_insensitive
	tol	0.000978	0.000043
	C	1.0	10
GBR	loss	absolute_error	absolute_error
	learning_rate	0.107452	0.037623
	n_estimators	194	476
	criterion	squared_error	friedman_mse
	min_samples_split	8	7
	min_samples_leaf	10	10
	max_depth	3	4
	max_features	1.0	sqrt
	n_iter_no_change	100	1000
ANN	Batch size	15	16
	Optimizer	Adam	RMSprop
	Initial learning rate	0.000444	0.000303
	Loss	huber_loss	huber_loss
	Learning rate scheduler	True	True
	Start epoch	12	12
	Learning rate decrease	0.822637	0.801706
	Every N epoch	8	7
	Early stopping	True	True
	Patience	7	9

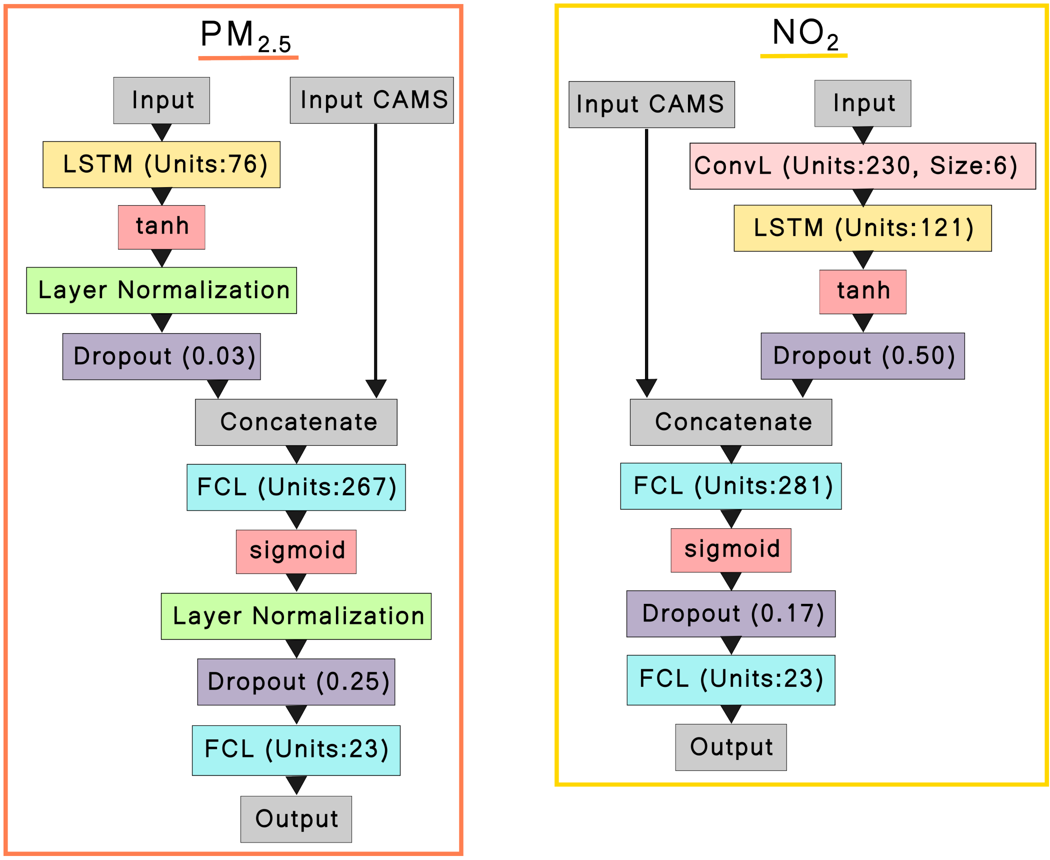


Figure 4.1: The ANN architectures found during the HPO are shown. The left side displays the architecture for predicting $PM_{2.5}$, and the right side displays the architecture used for predicting NO_2. Both networks are kept relatively simple concerning the number of layers and units per layer compared to the initial search space. Regularization techniques such as a dropout or normalization layer were used, and another fully connected layer (FCL) was applied after the concatenation. The right network added a 1D convolutional layer (ConvL) with a kernel size of six time steps as a feature extractor as the first layer.

Table 4.3: Descriptive statistics with regards to the MAE of each HP combination for the different ML algorithms, excluding the ANN are shown. Similar patterns for the minimum, maximum, mean, median MAE and the standard deviation can be seen for both pollutants for all regressors. The algorithms are sorted due to the number of optimized HPs in increasing order.

Algorithm	MAE $PM_{2.5}$					MAE NO_2				
	Min	Max	Mean	Median	Std	Min	Max	Mean	Median	Std
Ridge	3.17	5.09	3.32	3.18	0.28	5.90	8.40	6.13	5.90	0.38
Linear SVR	3.08	9.71	3.60	3.13	1.06	5.91	16.80	6.85	5.97	1.92
Lasso	3.15	5.42	3.77	3.16	0.94	5.80	9.31	6.74	5.82	1.36
GBR	3.07	8.26	3.64	3.13	0.88	5.55	15.82	6.55	5.71	1.50

deviation, and the GBR the lowest minimum and median MAE for both pollutants.

4.2. Scenario 1

With the found HPO presented in the previous section, this and the following sections deal with the results regarding the overall research question. More specifically, all evaluated ML algorithms were trained with the previously found HP to predict samples belonging to the time range of the test set (2022-06-01 to 2022-12-31) at all in-situ stations shown in Table 3.2. The corresponding results are compared against the CAMS prediction as the baseline. Table 4.4 depicts the overall performance of the different ML algorithms predicting the next 23 hours of $PM_{2.5}$ or NO_2 for S1. The performance was measured using the MAE.

Since the different learning algorithms' objective was to reduce the MAE, this metric is also used for the overall comparison of the results displayed in Table 4.4. The RMSE is only calculated and presented as a measure to compare against similar experiments from literature (Bertrand et al., 2023). Compared against CAMS, it can be seen that all locally employed algorithms outperformed the regional forecast. Generally, a higher percental improvement can be noticed for the NO_2 forecast than for $PM_{2.5}$. The lowest improvement for both target pollutants can be observed for the bias-corrected CAMS (CAMS BC) as employed by EEA, 2023. Additionally, a higher performance gain can be observed if CAMS was included as input for the different ML algorithms for $PM_{2.5}$ in contrast to NO_2. While the average improvement for NO_2 is below 5%, it is above 10% for $PM_{2.5}$. The overall improvement against CAMS varies from 13.05% (ANN) to the highest reduction of 19.32% (GBR) for predicting $PM_{2.5}$ while CAMS was not included as input. When CAMS served as additional input, the performance variation between the different ML algorithms is lower. The improvement ranges from 27.68% (Ridge) to the

Table 4.4: The table represents the results of S1 with $PM_{2.5}$ and NO_2 as target pollutants. It shows the average MAE and root mean squared error (RMSE) and R^2 of the regional forecast as baseline (CAMS) and compares all applied learning algorithms by providing the percental reduction towards CAMS. Furthermore, the prediction error of the locally bias-corrected CAMS prediction (CAMS BC) is shown. The highest percental reduction of the error with and without CAMS as input for the different ML algorithms is highlighted in bold. Comparable results from the literature are displayed at the bottom of the table.

	$PM_{2.5}$						$NO2$					
Algorithm	CAMS	MAE	RMSE	R^2	Reduction (%)		CAMS	MAE	RMSE	R^2	Reduction (%)	
					MAE	RMSE					MAE	RMSE
CAMS	-	3.83	6.17	0.379	0.0	0.0	-	9.89	14.21	-0.011	0.0	0.0
CAMS BC	-	3.72	5.98	0.422	2.87	3.08	-	7.53	10.56	0.41	23.76	25.68
ANN	No	3.33	5.51	0.501	13.05	10.70	No	6.34	9.1	0.585	35.89	35.96
	Yes	2.71	4.83	0.621	29.24	21.72	Yes	6.34	9.16	0.576	35.89	35.54
Lasso	No	3.18	5.23	0.547	16.97	15.24	No	6.29	8.74	0.622	36.40	38.49
	Yes	2.75	4.74	**0.632**	28.20	**23.18**	Yes	5.79	8.14	0.67	41.46	42.72
Ridge	No	3.18	5.2	**0.552**	16.97	**15.72**	No	6.35	8.75	0.621	35.79	38.42
	Yes	2.77	4.76	0.629	27.68	22.85	Yes	5.83	8.17	0.667	41.05	42.51
Linear SVR	No	3.13	5.33	0.527	18.28	13.61	No	6.21	8.84	0.614	37.21	37.79
	Yes	2.69	4.77	0.628	**29.77**	22.69	Yes	5.85	8.33	0.654	40.85	41.38
GBR	No	3.09	5.22	0.551	**19.32**	15.40	No	5.85	8.3	**0.659**	**40.85**	**41.59**
	Yes	2.76	4.87	0.613	27.94	21.07	Yes	5.44	7.77	**0.699**	**44.99**	**45.32**
Bertrand et al., 2023												
CAMS Literature	-	-	8.6	-	-	0.0	-	-	12.6	-	-	0.0
RF Literature	-	-	-	-	-	**22.00**	-	-	-	-	-	**33.00**

best-performing model for predicting $PM_{2.5}$ with 29.77% (Linear SVR). Shifting the focus on the prediction of NO_2 the lowest performance gain can be observed for the Ridge regression (35.79%) without CAMS as additional input. As for $PM_{2.5}$, the highest improvement without CAMS is achieved from the GBR (40.85%). If CAMS served as additional input, by far, the nethermost gain can be noted for the ANN (35.89%). Notably, the ANN's result is not improved over the local prediction without CAMS. In contrast, the highest performance gain is achieved again by the GBR (44.99%). Turning the attention to the R^2, it can be seen for $PM_{2.5}$ that the Lasso algorithm has the highest amount of explained variances in the observed data (0.632 or 63.2%) if CAMS was included and the similar Ridge regression the highest if no CAMS was incorporated as additional input (0.552). Both algorithms yield a higher R^2 compared to the CAMS prediction and explain 25.3% and 17.3% more of the variability in the observed data, respectively. By shifting the focus on the R^2 of the NO_2 predictions, the negative R^2 for the CAMS prediction is most notable. This results from the fact that the prediction is, on average, worse than the case if the mean value of the observations had been predicted for all samples. Even though the performance gain in explained variability regarding the CAMS prediction is higher than for the other metrics, the highest gain is also achieved by the GBR. Without CAMS as additional input, the GBR can explain 0.659 of the variability of the observed data. Similarly to the other metrics, this value increases by a small margin when MOS is applied (0.699). Overall, the coefficient of determination reflects the performance of the different ML algorithms if compared based on the MAE and RMSE. In comparison to the results achieved by Bertrand et al., 2023, it can be seen that for predicting $PM_{2.5}$, a similar performance can be observed for RMSE with 23.18% in this research and 22.00% in Bertrand et al., 2023. A higher difference can be noticed for NO_2. Here, the percental reduction for RMSE against CAMS in this research is 45.32% compared to 33.00% stated by Bertrand et al., 2023. It should be noted that in both cases ($PM_{2.5}$ and NO_2), the underlying data set is not the same. While this study focuses on urban environments, Bertrand et al., 2023 includes stations located in the country site, as well. The following subsections unfold a more detailed view of the results of predicting the two target pollutants.

4.2.1. $PM_{2.5}$

Focusing on the prediction of $PM_{2.5}$ Figure 4.2 shows the distribution of the MAE in predicting the next 23 hours per station and algorithm displayed as a boxplot. Compared to the overall mean MAE shown Table 4.4, Figure 4.2 shows similar patterns. All ML algorithms have a lower median MAE than CAMS and CAMS_BC, which is further reduced when CAMS is included as input to the learning algorithms (blue boxes). Figure 4.2 additionally shows the spread of the

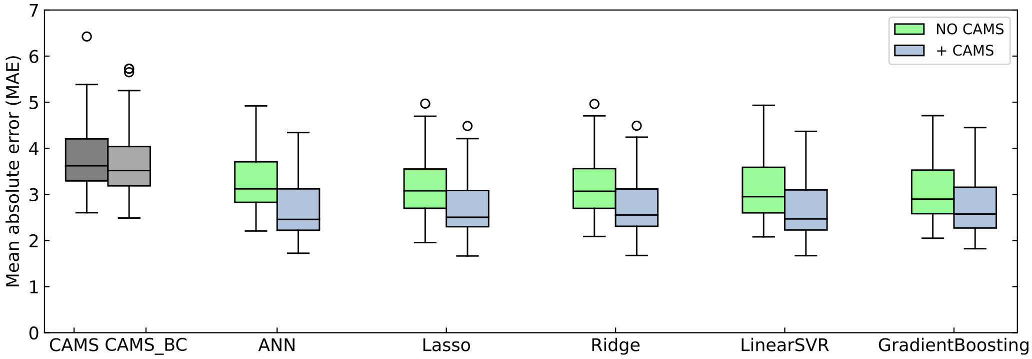


Figure 4.2: Comparison of different regressors predicting $PM_{2.5}$. The presented results reflect the results given in Table 4.4 so that based on the median lines, the central tendency of all employed ML algorithms is lower than for the regional forecasts (CAMS, CAMS_BC). Even though the spread of the predictions has a similar range between the whiskers, a clear spread shift can be noticed from the regional forecast to the ML algorithms predicting $PM_{2.5}$ without incorporating CAMS (green boxes). Remarkably, all ML algorithms can reduce not only the median line while incorporating CAMS as input (blue boxes) but also the spread of data points that lay between the median line and the lower bound of the boxes (Q1).

MAE between measured and predicted values at each station.

It is notable that while the spread around the median line follows similar patterns in all predictions (the distance between the median line and the upper whisker is higher than to the lower whisker, showing a higher divergence from the median line for higher MAE), the overall spread is shifted around the reduced median MAE of the different ML algorithms. Interestingly, when CAMS is included as input, not only the median MAE is lowered, but also the spread of data points that lie between the median line and the lower bound of the boxes (Q1) is reduced. Figure 4.3 shows the average prediction error per hour, comparing the different local ML against the regional CAMS predictions when no MOS is applied. The best-performing algorithms from Table 4.4 are presented, and the ANN is additionally included as the most complex model. It can be seen that both local ML algorithms show similar patterns in contrast to the regional CAMS forecast. While the regional forecast yields a relatively stable (higher) MAE over the day, for the two local ML approaches, the average MAE is increasing over the day, leading to the highest error at the end of the horizon (20 to 23 hours ahead). While the difference of the average MAE at prediction time step t+1 (1 a.m.) between the local approaches and regional prediction is highest (approximately 2 $\mu g/m^3$ $PM_{2.5}$), it gradually decreases to a similar error at 11 a.m., staying similar for the rest of the day. Regarding the MSD, both local ML algorithms are first overestimating the $PM_{2.5}$

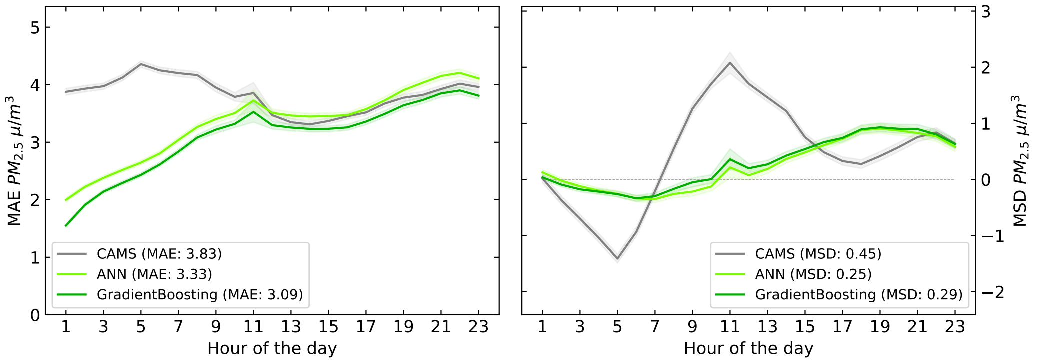


Figure 4.3: Displayed is the average MAE on the left and the mean signed deviation (MSD) on the right per hour of the day for the CAMS and local ML predictions. No MOS was applied. While ANN and GBR show similar patterns, the average error of CAMS diverges. For the MAE, the highest difference can be observed before 11 a.m. Additionally, the CAMS prediction shows a higher amplitude for its peaks at 5 a.m. and 11 a.m.

concentration before 10 a.m. and then underestimating the measures after 10 a.m. A higher MSD can be observed for the CAMS prediction. It shows a higher amplitude of the peaks at 5 a.m. (overestimation) and 11 a.m. (underestimating). Notably, all predictions show at least a small peak at 11 a.m. for the MAE and MSD. While Figure 4.3 compared the average hourly error between the local and regional approaches without applying MOS, Figure 4.4 compares the average hourly error when MOS is utilized. The prediction errors for the ANN, GBR, CAMS, and bias-corrected CAMS (CAMS_BC) are shown. Even though a high resemblance can be observed between the locally applied ML algorithms, some differences can be seen. Most importantly, an overall reduction of the average MAE of the ML algorithms leads to an in Figure 4.3 previously unseen gap between the CAMS prediction error and the local ML learning prediction errors after 11 a.m. The overall MAE for the ANN is reduced by 0.62 from 3.33 to 2.71, and the MAE for the Linear SVR by 0.4 from 3.09 to 2.69. Apart from the overall reduction of the average MAE, the small but continuously higher MAE for the ANN compared to the SVR disappears after 3 a.m. if CAMS serves as additional input.

Interestingly, even though the applied method to bias correct CAMS lowers the MSD close to zero, CAMS_BC is only yielding a slight reduction of the MAE compared to the raw CAMS prediction. Figure 4.5 shows a sample prediction of one week for a station in Frankfurt. At midnight, the different models predict the $PM_{2.5}$ concentration for the next 23 hours, incorporating one or more past time steps. Seven predictions (highlighted as dashed vertical lines) are performed sequentially in total.

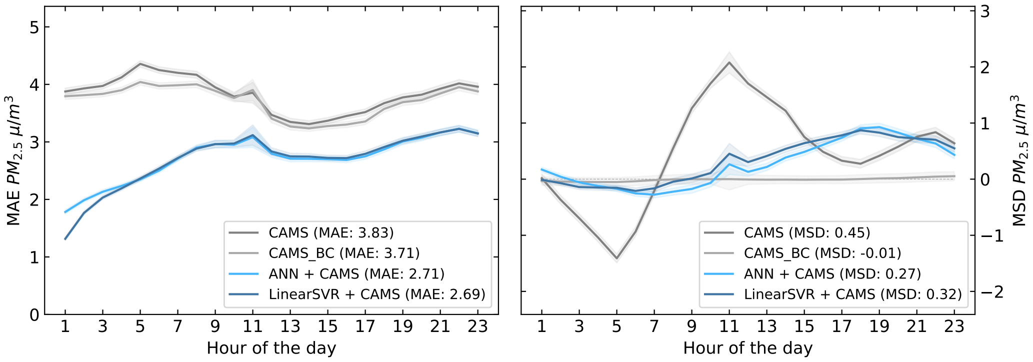


Figure 4.4: As in Figure 4.3, the hourly average MAE and MSD of the predictions are displayed on the left and right, respectively. In contrast, MOS is applied. The locally bias-corrected CAMS prediction (CAMS_BC) is also displayed. There is a high pattern resemblance between the local prediction errors compared to Figure 4.3 with a few exceptions. Additionally, an overall reduction of the average MAE can be seen for the locally applied ML algorithms, which leads to a previously in Figure 4.3 unseen gap between the CAMS prediction error and the local ML learning approaches after 11 a.m.

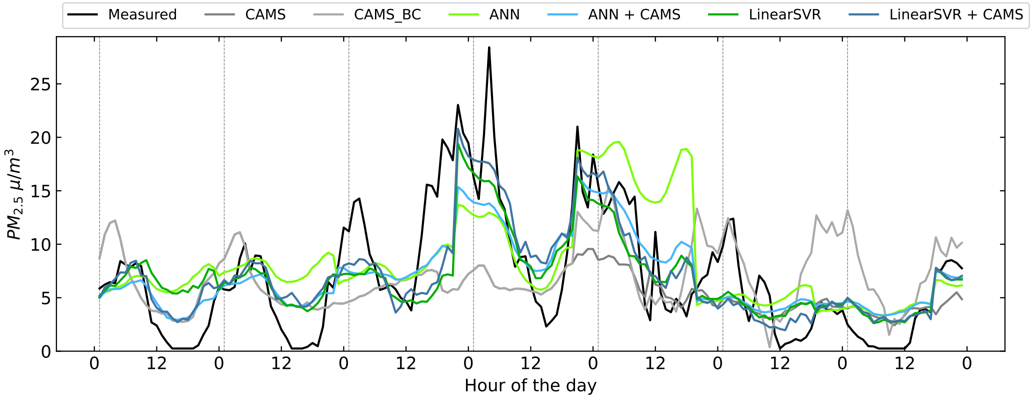


Figure 4.5: A one-week example prediction for a station in Frankfurt is shown. The model with the lowest MAE (Linear SVR) and most complex (ANN) are displayed. The different predictions are compared against the measured data. “Model forecast [t+1]” refers to the vertical dashed lines and indicates the first prediction of the different models for the time step t+1. It can be seen that, especially at midnight, high peaks of measured $PM_{2.5}$ concentration are captured more accurately by the local ML prediction, which reflects the patterns shown in Figure 4.3 and Figure 4.4.

The sample prediction underlines the patterns seen in Figure 4.3 and Figure 4.4. During the first hours of the day, the local ML predictions are more capable of following the peak concentrations than the regional forecast CAMS. Except for the

ANN, not including the CAMS prediction, all locally applied ML algorithms show similar patterns to each other. While it is challenging to capture sudden measured concentration peaks or drops, generally, they are more often captured during the first hours of the predictions. Notably, sudden concentration drops in the afternoon are often not captured by any prediction. Figure 4.6 shows a confusion matrix of the predicted and true labels if $PM_{2.5}$ concentration is categorized into the proposed air quality indices shown in Table 1.1. The assignment was performed for each hour. Note that the employed algorithms were not trained to perform a classification task. Instead, the predicted and measured concentration values were subsequently assigned to the proposed categories. From Figure 4.6, it can be seen that concentration levels that appear rarely are also detected relatively seldom in comparison to the classes represented more often. Still, the class indices “Very poor” and “Extremely poor” are captured more often by the SVR. While the Linear SVR detects 17.3% and 4.85% percent of the “Very poor” and “Extremely poor” labeled classes, the CAMS prediction only detects 5.72% and 0%, respectively. Generally, the regional forecast underestimates the measured $PM_{2.5}$ concentration more often compared to the Linear SVR. After the predictions of $PM_{2.5}$ concentration were elaborated in more detail, the focus shifts to the estimation of NO_2 concentration in the following section.

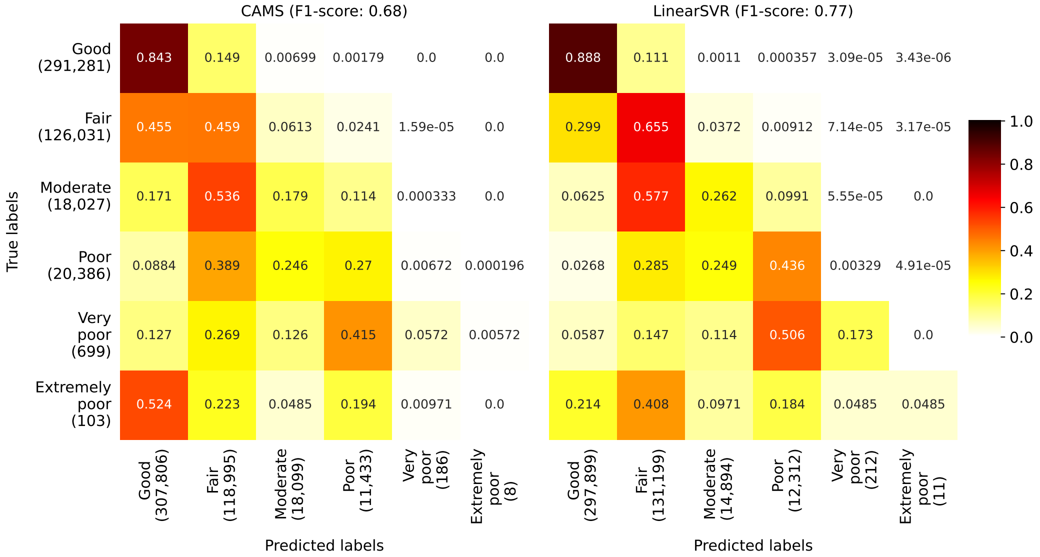


Figure 4.6: Comparison of the classification into the different air quality indices defined by the European Environmental Agency (EEA) and shown in Table 1.1. The number of predicted hours per class is stated in brackets, and the annotations show the relative number of samples. It can be seen that the Linear SVR, as algorithm with the lowest MAE, has slightly but continuously higher true positives for all classes.

4.2.2. $NO2$

Figure 4.7 shows analogical to Figure 4.2 the distribution of the prediction MAE over all elaborated stations measuring NO_2. It compares the MAE of the different regressors predicting NO_2 for the elaborated stations. CAMS and CAMS_BC represent the regional forecast, whereas the latter is bias-corrected to incorporate the past local pollutant measurements. Similar to Figure 4.2, it can be seen that all learning algorithms that utilized local measurements outperform the CAMS prediction (green boxes). Moreover, all algorithms, except for the ANN, decrease the median MAE further while CAMS was incorporated. The relative reduction is lower compared to Figure 4.2, which is also reflected through the average MAE shown in Table 4.4. The CAMS prediction has a relatively higher spread compared to the other regressors, in contrast to Figure 4.2. Additionally, the relative reduction of CAMS_BC towards CAMS is more distinct. Furthermore, the number of represented outliers above the median line increases except for the Ridge regression. Following the overall MAE and different error distributions, Figure 4.8 shows the average errors per hour and algorithm. The displayed local ML algorithm did not include the CAMS prediction as additional input.

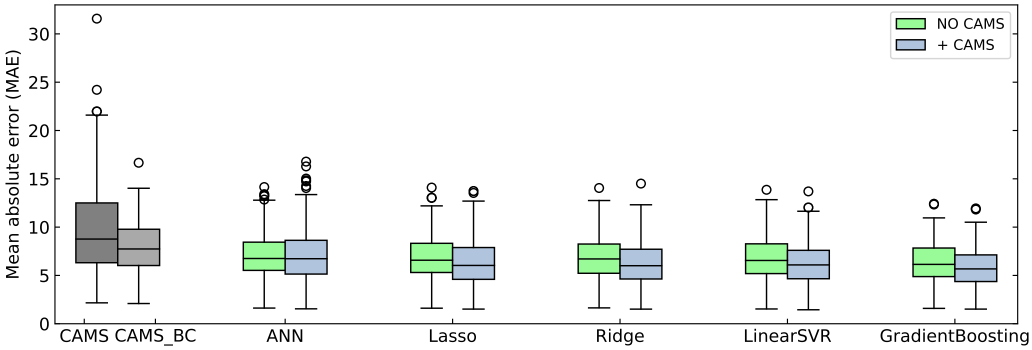


Figure 4.7: The box plots outline the MAE distribution of the employed regressors for predicting NO_2 at each target station. Similar to Figure 4.2, all locally applied ML algorithms outperform the regional forecasts, with an additional relatively small reduction of the median MAE (except for the ANN), when CAMS served as additional input.

Even though all presented algorithms reveal a similar pattern, a clear gap between the locally applied predictions and the regional forecast can be observed, with the highest error for the CAMS prediction between 7 a.m. and 9 a.m. and for the local predictions towards the end of the day. While the MSD for the ANN and GBR are relatively stable throughout the day compared to the CAMS prediction, the latter depicts a relatively high underestimation, with its average peak of 10 $\mu g/m^3$ NO_2 concentration at 9 a.m. The lower average prediction MAE from

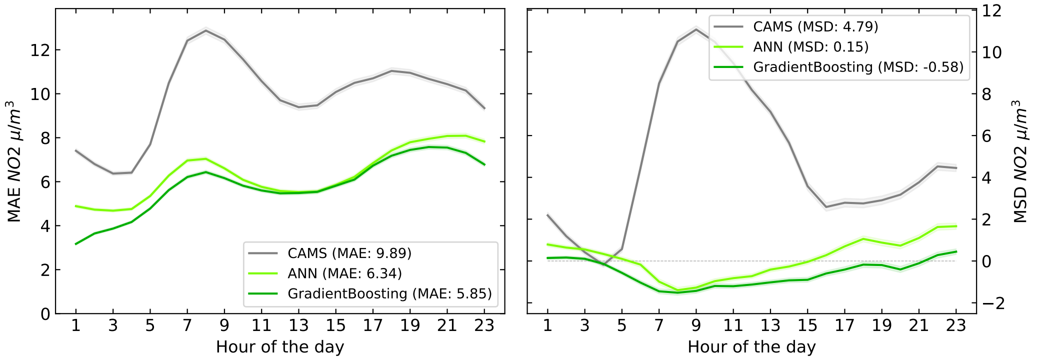


Figure 4.8: Similarly to Figure 4.3, the average hourly MAE (left) and MSD (right) were calculated based on the NO_2 predictions for all evaluated measurement stations, without incorporating MOS. Even though all MAE follow a similar pattern, a clear gap can be observed between the local ML and regional CAMS predictions. Moreover, the latter shows a relatively high underestimation (reflected by the MSD) compared to the local predictions.

the GBR, as already shown in Table 4.4, results from a slightly more accurate prediction before 10 a.m. and after 17 o'clock against the predictions achieved by the ANN. The prediction MAE in the hours between is nearly identical. As for $PM_{2.5}$ Figure 4.9 displays the average hourly prediction errors for predicting the NO_2 concentration when CAMS was incorporated.

The average hourly MAE and MSD for the regional forecast CAMS, the bias-corrected CAMS (CAMS_BC) and the locally applied ML algorithms ANN and GBR are displayed. While the average MAE for the ANN is not changing, the GBR can further lower the average MAE, leading to a wider gap between the two ML algorithms. Even though the MSD for both algorithms is relatively stable compared to the CAMS prediction, a slight underestimation ($\sim$2 $\mu g/m^3$) between 7 a.m. and 9 a.m. can be noticed. As in Figure 4.4, the bias of the CAMS prediction reflected with the MSD per hour can be reduced to close to zero by employing the CAMS_BC prediction. In contrast, though, this leads to a relatively higher reduction for the MAE of the NO_2 prediction (23.76%) compared to $PM_{2.5}$ (2.87%), as can be seen in Table 4.4. After the overall performance of the different models was evaluated in general and with regards to the MAE and the MSD, Figure 4.10 depicts a one-week sample prediction for a station in Frankfurt.

Figure 4.10 shows an exemplary prediction for a single station inside Frankfurt. Even though the applied ML algorithms often capture concentration peaks in the first half of the day accurately, a second peak on the third day is not predicted by any model. Nevertheless, it is interesting to see that the locally employed ML algorithms anticipate a concentration raise at the end of the day. Furthermore, it

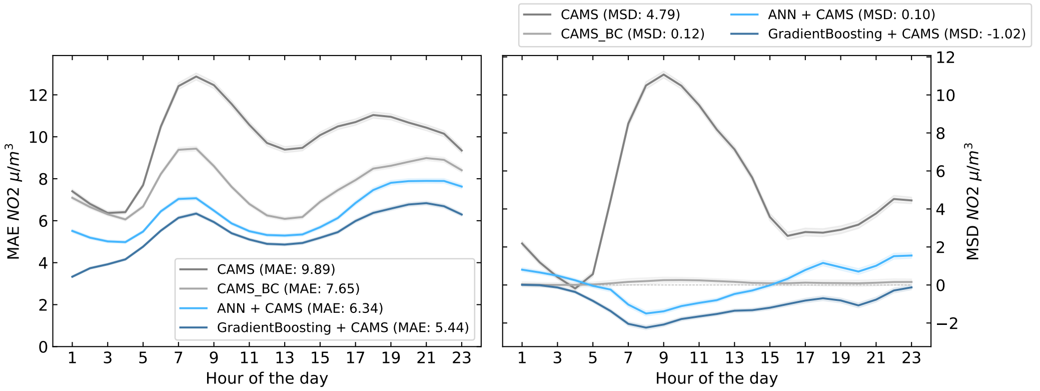


Figure 4.9: As in Figure 4.8 the average hourly MAE (left) and MSD (right) is presented, employing MOS. While MSD of the ANN and the GBR are relatively stabilized, the latter can further decrease the MAE. The bias-corrected CAMS (CAMS_BC) is also displayed. Similarly, the hourly bias indicated on the right can be reduced to close to zero. In contrast, the impact of the relative decrease on the MAE is higher, as can be seen in Table 4.4.

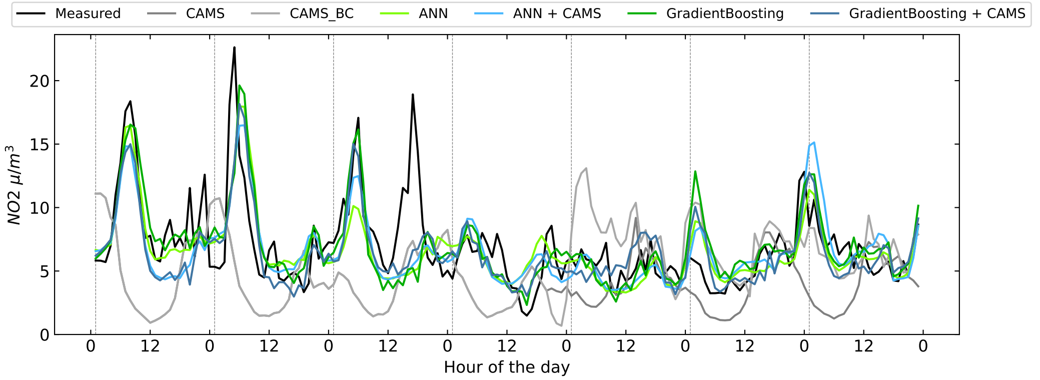


Figure 4.10: A one-week example prediction for a station in Frankfurt is shown. The model with the lowest MAE (GBR) and most complex (ANN) are displayed. The different predictions are compared against the measured data. "Model forecast [t+1]" refers to the vertical dashed lines and indicates the first prediction of the different models for the time step t+1. It can be seen that especially during the first half of the day, high peaks of measured NO_2 concentration are captured more accurately by the local ML prediction, which reflects the patterns shown in Figure 4.8 and Figure 4.9.

can be seen that in this one-week sample, the CAMS_BC prediction underestimates the measured prediction more often than the original regional prediction CAMS. Figure 4.11 displays, similar to Figure 4.6, a confusion matrix of the predicted and true labels, if the NO_2 concentration is categorized into the air quality indices shown in Table 1.1. Because the categories "Very poor" and "Extremely poor" are neither measured nor predicted for NO_2, they are not displayed in Figure 4.11 in contrast to Figure 4.6. A comparison between the CAMS regional prediction and the GBR with the lowest MAE in Table 4.4 is presented. While the CAMS prediction is slightly more often able to predict the concentration level labeled as "Good" (+ 0.8%), the GBR detects the classes "Fair" and "Moderate" more accurately (+36.1% and +2%, respectively). Notably, the GBR more often predicts a neighboring class, as in the true label as "Moderate" and the predicted as "Fair". Furthermore, neither of the algorithms can detect the measured 35 hours labeled as "Poor".

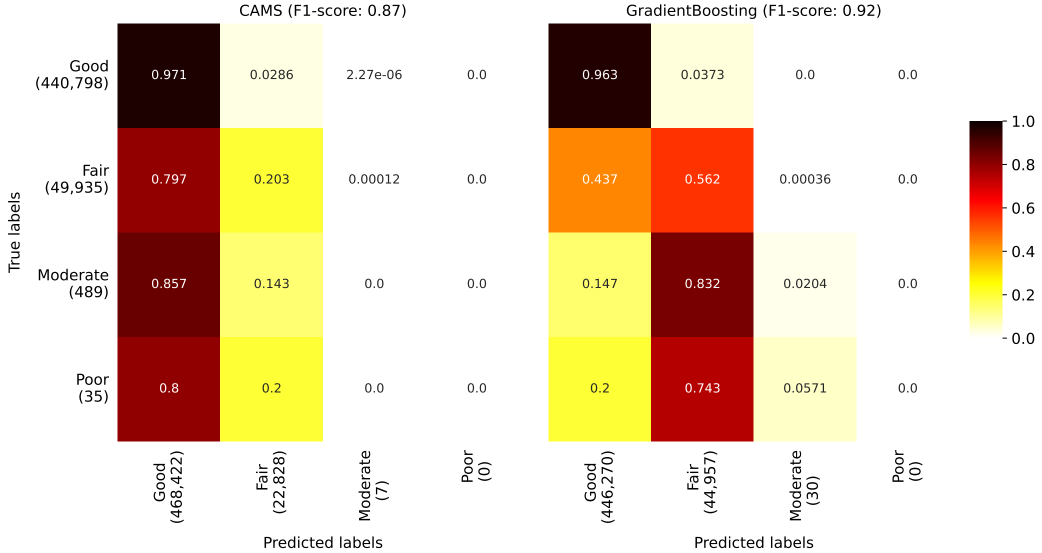


Figure 4.11: Comparison of the classification into the different air quality indices defined by the EEA and shown in Table 1.1 are depicted. The number of predicted hours per class is stated in brackets, and the annotations show the relative amount of samples. It can be seen that while CAMS predicts the class "Good" more accurately, the GBR detects higher values more often.

4.3. Scenario 2

After stating the results for S1 to answer the research question, if regional forecast can be improved locally in an urban environment using ML, this section presents the results for S2, in which it will be elaborated if the previously achieved performance of S1 can be improved by incorporating the information of neighboring stations. In S2, the neighboring stations were additionally trained to predict the pollutant concentration at the target station. The different predictions at one location were averaged for each hour. A more detailed description of the different scenarios can be found in Section 3.3.4. Table 4.5 depicts a comparison against the regional forecast CAMS and if each locally applied algorithm for S1 can be improved.

Table 4.5: A comparison between the results achieved for S2 in comparison to the regional CAMS prediction and the results achieved for the experiments of S1 is presented. For each algorithm employed, the percental difference is stated. Even though all algorithms yield a lower MAE than the CAMS prediction for $PM_{2.5}$ and NO_2, only a slight reduction can be observed for most of the algorithms predicting $PM_{2.5}$.

	$PM_{2.5}$				$NO2$			
Algorithm	CAMS	MAE	Reduction (%)		CAMS	MAE	Reduction (%)	
			CAMS	S1			CAMS	S1
CAMS	-	3.83	-	-	-	9.89	-	-
ANN	No	3.30	13.84	0.90	No	7.13	27.91	-12.46
	Yes	2.87	26.07	-5.58	Yes	6.76	31.65	-6.62
Lasso	No	3.13	18.28	1.57	No	6.41	35.19	-1.91
	Yes	2.73	28.72	0.73	Yes	5.95	39.84	-2.76
Ridge	No	3.08	19.58	**3.14**	No	6.48	34.48	-2.05
	Yes	2.76	27.94	0.36	Yes	5.92	40.14	**-1.54**
LinearSVR	No	3.18	16.97	-1.60	No	6.51	34.18	-4.83
	Yes	2.67	**30.29**	0.74	Yes	6.02	39.13	-2.91
GBR	No	3.03	**20.89**	1.94	No	5.92	**40.14**	**-1.20**
	Yes	2.71	29.24	**1.81**	Yes	5.59	**43.47**	-2.57

It can be seen that while all algorithms can reduce the MAE in comparison with CAMS when predicting $PM_{2.5}$ and NO_2, only a slight reduction can be observed for the majority of algorithms predicting $PM_{2.5}$ if compared against the predictions of S1 stated in Table 4.4. Indeed, for NO_2, only an increase of the MAE can be seen when compared against the S1 prediction. Noticeably, the ANN has the highest increase in MAE compared against S1, with 12.46% and 6.62% above the MAE with and without CAMS as input, respectively. It should be highlighted that the HP shown in Table 4.2 and Figure 4.1 were not used in nor optimized for

S2. Figure 4.12 compares the predictions for S1 and S2 concerning the ANN and CAMS, if no MOS is performed.

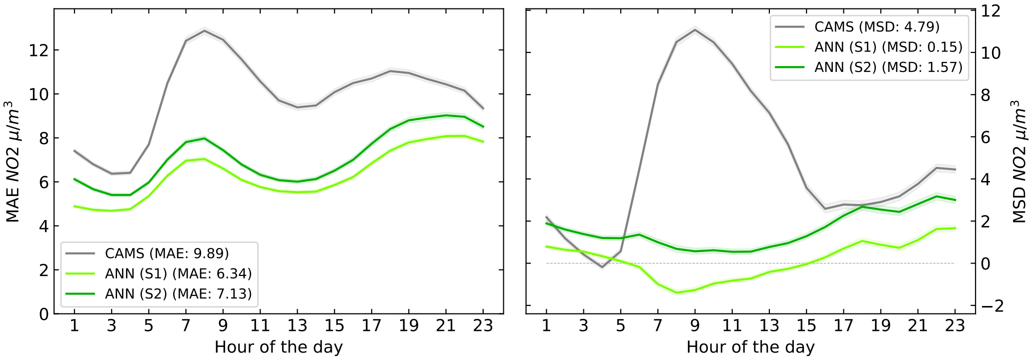


Figure 4.12: Comparison of the ANN trained for S1 and S2 (shown in the brackets) without MOS. The average hourly MAE and MSD are displayed on the left and right, respectively. While both ANN predictions have a lower MAE than CAMS throughout the day, the results for S2 are also continuously lower than the ones achieved for S1.

The ANN for the two scenarios can continuously lower the average MAE per hour compared to CAMS. Even though the ANN predictions follow similar patterns for MAE and MSD, a slight increase in MAE can be observed if the neighboring stations are included (S2). Also, an increase for the MSD resulting in a slight pattern shift can be noticed. Figure 4.13 compares the predictions for S1 and S2 with respect to the ANN and CAMS, while MOS was applied.

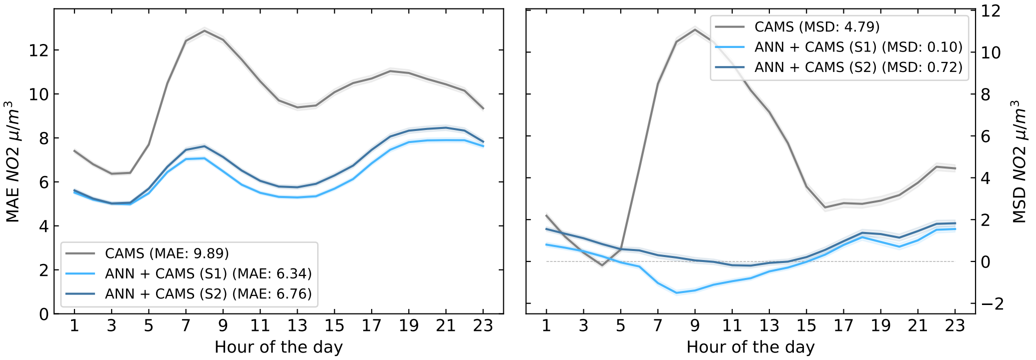


Figure 4.13: Comparison of the ANN trained for S1 and S2 (shown in the brackets) with MOS. The average hourly MAE and MSD are displayed on the left and right, respectively. Even though the ANN employing neighboring stations in S2 still achieves an overall higher MAE compared to S1, the difference is not as distinct as without MOS.

As in Figure 4.12, both ANN achieve a lower MAE as the regional forecast CAMS. In contrast, the ANN predictions for S2 show no apparent difference in the first hours of the day. Furthermore, the overall introduced bias while neighboring stations were incorporated is lower than for Figure 4.12. After the description of S2, the focus is shifted to S3, where the historical pollutant measurements at the target stations were excluded.

4.4. Scenario 3

This section presents the results of the different ML algorithms for S3. S3 dealt with the question how accurate the target pollutant concentration can be estimated at a target point in an urban environment, when neighboring stations are included. The neighbors were combined by averaging the prediction of each station at the target location. A more detailed description can be found in Section 3.3.4. Furthermore, the experiment was performed with and without the regional forecast CAMS as additional input to investigate the impact on the predictions. Table 4.6 presents the results compared to the CAMS forecast.

Noticeably, except for XGBoost_OM, all ML algorithms can reduce the overall MAE compared to the regional forecast CAMS by incorporating CAMS as additional input. The highest reduction against the CAMS prediction MAE was achieved by the GBR, with 15.14% for $PM_{2.5}$ and 18.71% for NO_2. While this shows similar results compared between the two target pollutants, more distinct differences can be observed if CAMS did not serve as additional input. While for $PM_{2.5}$, all employed ML algorithms achieve a lower MAE compared to CAMS, predicting NO_2 from neighboring stations often yields a higher MAE. More specifically, the highest MAE reduction against the CAMS only using neighboring stations is achieved by the ANN with 10.18% compared to the highest reduction of NO_2 against CAMS for the GBR with only 1.82%.

The specific cases for predicting the two pollutants at a target location are shown in the lower part of Table 4.6. They include the GBR incorporating the distance to the target point (GBR_Dist) and the XGB utilizing each of the neighboring stations simultaneously (XGBoost_OM). Since there was no HP tuning performed for the XGB in any scenario, the average prediction as for the other algorithms is also stated with default HP settings for the XGB (XGBoost). While investigating the specific cases, it is evident that the employed ML algorithms for $PM_{2.5}$ also reduce the MAE compared to CAMS, even though the MAE is higher in contrast with the associated counterparts GBR for GBR_Dist and XGBoost for XGBoost_OM. After establishing the overall results for S3, the two target pollutants are investigated in more detail in the following subsections.

Table 4.6: The results achieved during the experiments concerning S3 are presented. Differences can be observed in the results for $PM_{2.5}$ and NO_2. While the prediction of both pollutants yields a lower MAE than CAMS if MOS is applied (except for XGBoost_OM), only the ML algorithms predicting $PM_{2.5}$ achieve a reduced MAE compared to CAMS. The lower part of the table shows the GBR when the distance to the target is included (GBR_Dist) and the extreme gradient boosting (XGB) algorithms, including all neighboring stations simultaneously (XGBoost_OM).

	$PM_{2.5}$			$NO2$		
Algorithm	CAMS	MAE	Reduction (%)	CAMS	MAE	Reduction (%)
CAMS	-	3.83	0.0	-	9.89	0.0
ANN	No	3.44	**10.18**	No	9.79	1.01
	Yes	3.42	10.70	Yes	8.38	15.27
Lasso	No	3.62	5.48	No	10.31	-4.25
	Yes	3.35	12.53	Yes	9.19	7.08
Ridge	No	3.55	7.31	No	10.30	-4.15
	Yes	3.35	12.53	Yes	9.14	7.58
LinearSVR	No	3.47	9.40	No	9.80	0.91
	Yes	3.27	14.62	Yes	8.90	10.01
GBR	No	3.47	9.40	No	9.71	**1.82**
	Yes	3.25	**15.14**	Yes	8.04	**18.71**
GBR_Dist	No	3.80	0.78	No	9.71	1.82
	Yes	3.61	5.74	Yes	8.37	15.37
XGBoost	No	3.61	5.74	No	10.28	-3.94
	Yes	3.43	10.44	Yes	8.87	10.31
XGBoost_OM	No	3.74	2.35	No	11.44	-15.67
	Yes	3.51	8.36	Yes	11.71	-18.40

4.4.1. $PM_{2.5}$

Figure 4.14 depicts the distribution of the MAE, comparing the CAMS predictions against the locally applied ML algorithms for $PM_{2.5}$ for S3.

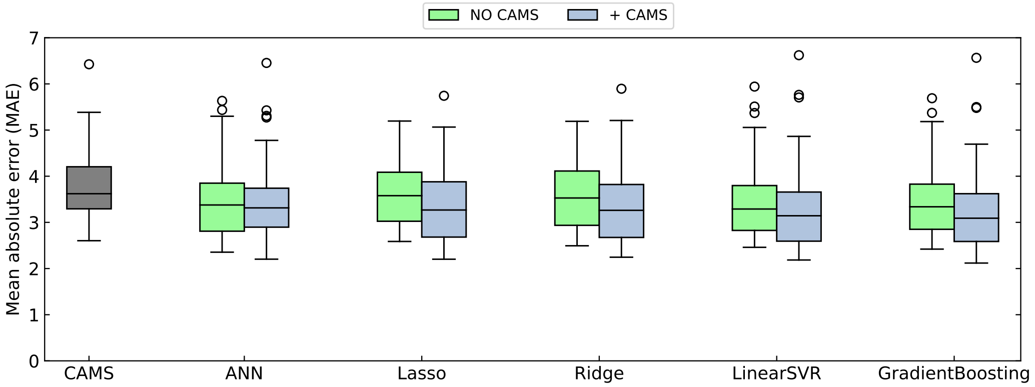


Figure 4.14: The distributions of the MAE are shown for predicting all elaborated target points for each applied algorithm. While all locally employed algorithms median line is lower compared to CAMS and all ML algorithms can reduce the median MAE per target point further if CAMS is included as additional input, for the ANN, Linear SVR and GBR outliers can be observed, for which an increased MAE was achieved.

Similarly to Figure 4.2, all median lines of the locally applied algorithms yield a lower MAE in predicting $PM_{2.5}$ compared to CAMS. As Table 4.6 suggests, the reduction is less distinct compared to S1. While an additional reduction in the median station MAE was achieved by incorporating the CAMS prediction for nearly all algorithms, a higher amount of outliers above the median were also induced, some even extending the outlier in the CAMS prediction (ANN, SVR, GBR). Even though there are no clear patterns of the data distribution visible across all employed algorithms, some algorithms narrowing the spread if CAMS is included as additional input (ANN), other algorithms do not show a comparable change. In general, though, it can be seen that the data distribution above the median has a higher spread than below.

After showing the distribution of the MAE resulting from the predictions per target point and algorithms, Figure 4.15 shows the average error per hour of the day for all target points predicted in S3. No MOS was applied in this setup. Similarly to the results achieved for S1, the ANN and GBR particularly yield a lower but less distinct average MAE before 11 a.m. compared to the CAMS forecast. In contrast, the CAMS prediction returns a slightly lower MAE after 11 a.m. for the rest of the day. For the MSD, the GBR and the ANN show nearly identical patterns compared to S1, except that for the latter, a slight shift to an overall

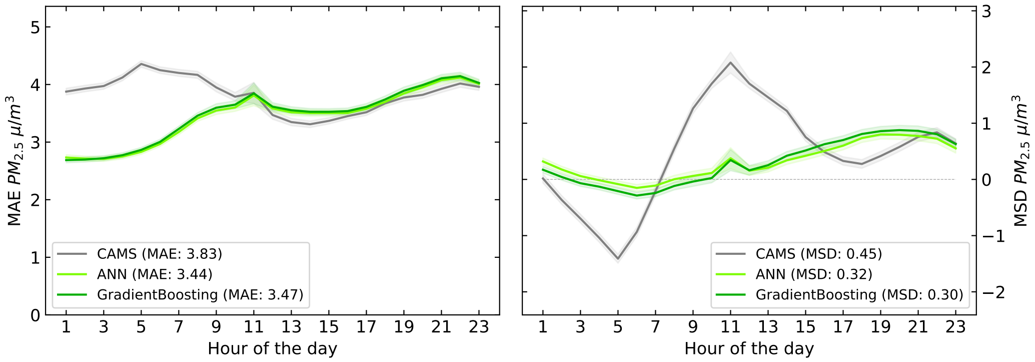


Figure 4.15: The average MAE on the left and the MSD on the right per hour of the day for the CAMS and local ML predictions is displayed. No MOS was applied. The lower overall MAE for the ANN and GBR compared to CAMS was achieved during the first half of the day. For the MSD, the GBR and the ANN show nearly identical patterns compared to S1, except that for the latter, a slight shift to an overall increased MSD can be noticed.

increased MSD can be noticed. Figure 4.16 presents the average hourly prediction errors when MOS was applied. It can be seen that while the ANN was only able to

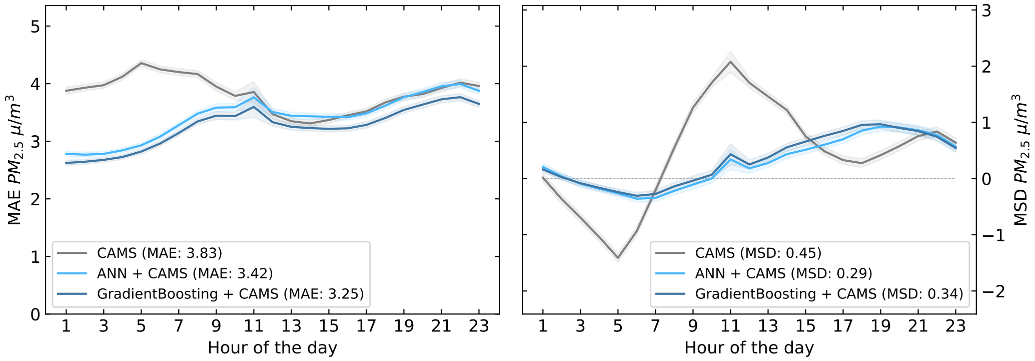


Figure 4.16: The average hourly MAE (left) and MSD (right) if no MOS is applied are shown. While a negligible reduction in MAE can be noticed for the ANN, the GBR is capable of slightly reducing the average MAE further throughout the day compared to the ANN and CAMS, even though a slight increase can be noticed in MSD in comparison when no MOS is applied.

negligible decrease the MAE if CAMS was included, the GBR is able to reduce the MAE slightly. On the other hand, the overall MSD is reduced by the ANN, whereas a slight increase can be observed for the GBR compared to Figure 4.15.

Comparing S3 with the results from S1 (Figure 4.4) for the prediction of $PM_{2.5}$, it can be seen that the average hourly error patterns are similar. In particular, if no MOS was applied, both scenarios show that the overall lower MAE is achieved in the first half of the day, with a higher reduction if local measurements are included as in S1. The second half only shows slight differences, with the ANN achieving a continuously higher MAE in both scenarios if compared against CAMS and the GBR a slightly lower and higher MAE compared to CAMS for S1 and S3, respectively. For the MSD, the differences between S3 and S1 are even less distinct. A different picture can be seen when comparing S3 and S1 if MOS was applied. As can be seen from Figure 4.16 (S3) in comparison with Figure 4.4 (S1), the algorithms employed in S3 can not reduce the average MAE in particular of the second half of the day as much as in S1. Surprisingly, the MSD does not show this distinct change while comparing S3 to S1. After presenting the results for the prediction of $PM_{2.5}$ in the context of S3 and comparing them against S1, the following section shows the results for predicting NO_2.

4.4.2. NO_2

Figure 4.17 shows the distribution of the MAE for predicting NO_2 at the evaluated target points. Noticeably, the spread of 50% of the data points around the median line (boxes) can be narrowed by all locally applied algorithms, in particular when CAMS was included as additional input. Even though an apparent reduction of the median MAE over all stations can only be seen for the GBR when

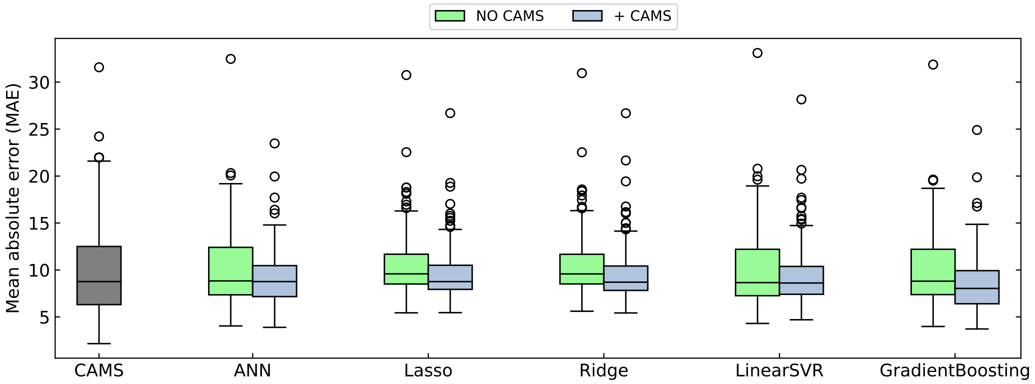


Figure 4.17: The average MAE per evaluated station and different algorithms are presented. The regional CAMS forecast is compared against the locally applied algorithm, either without or with CAMS as additional input. While all boxes of locally employed algorithms show a smaller spread, only the GBR clearly show a reduced median compared to CAMS.

CAMS was included if compared against CAMS. Figure 4.18 shows the average hourly prediction errors for NO_2 when no MOS was applied.

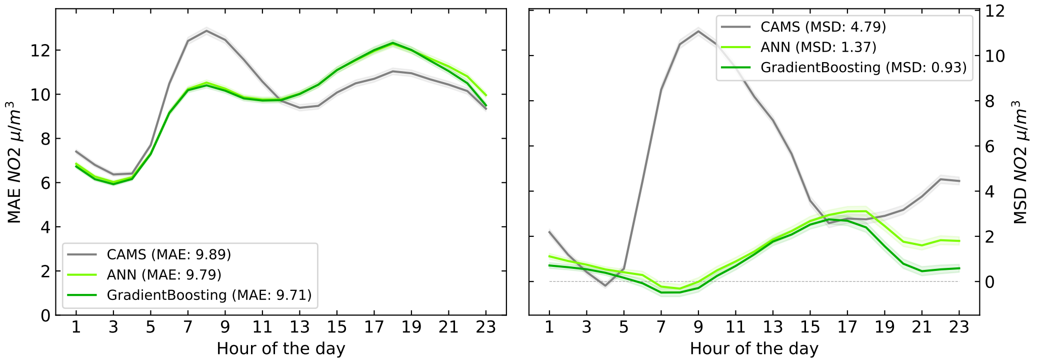


Figure 4.18: The average hourly MAE (left) and MSD (right) if no MOS was applied are shown. It can be seen that while the ANN and GBR achieve a nearly identical MAE throughout the day, they yield a lower MAE compared to CAMS, particularly in first hours of the day, with the highest difference between 6 a.m. and 12 a.m. and a higher in the second. Both algorithms also show an increased underestimation of the measured pollutants in the second half of the day.

Both locally applied ML algorithms show nearly identical error patterns for MAE throughout the day. While the average MAE per hour is lower in the first half of the day compared to CAMS when averaging over the predictions of the neighboring stations (S3), the same can not be seen for the second half. After 12 a.m., the average MAE shows a clear upward trend, peaking at 18 o'clock. Compared with CAMS, the gap closes again towards the end of the day. Turning attention to the MSD, it can be seen that the peak underestimation of CAMS at 9 a.m. is not mirrored in either of the locally applied ML algorithms. These show an increased underestimation after 11 a.m. throughout the day, peaking at 16 o'clock for the GBR and 18 o'clock for the ANN. Figure 4.19 similarly shows the average hourly prediction errors for the regional CAMS forecast and the predictions of the locally applied algorithms ANN and GBR, but using the CAMS prediction as additional input. Similarly to S1 and S2, the average hourly MAE shows an overall upward trend towards the end of the day, with two peaks at 9 a.m. and 19 o'clock. Also, an apparent reduction in MAE towards CAMS if compared against Figure 4.18 when no MOS was applied can be seen. While the ANN and the GBR show an apparent reduction in MAE compared to CAMS throughout the day, the GBR's reduction is slightly higher. Interestingly, including CAMS as additional input to the locally applied ML algorithms is nearly neglecting the previously existing bias as present in Figure 4.18.

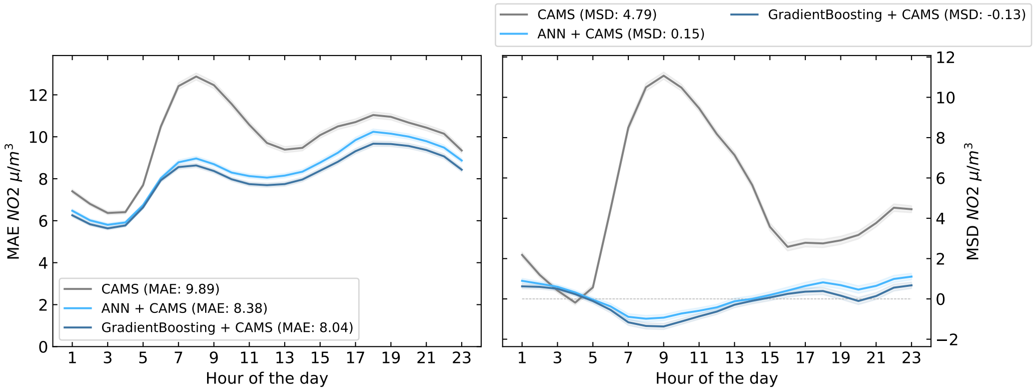


Figure 4.19: The average hourly MAE (left) and MSD (right) while incorporating MOS are shown. It can be seen that the ANN and GBR achieved a continuously lower MAE throughout the day. Furthermore, the previously existing bias (represented by the MSD) for both algorithms could be reduced to close to zero when compared to Figure 4.18.

4.5. Synopsis

First, the results of the HPO for different ML algorithms are presented in Section 4.1. For each algorithm, Table 4.1 shows the identified input features and past time steps included in the prediction of $PM_{2.5}$ and NO_2. Similarities and differences for the HP configuration of the different ML algorithms concerning the target pollutants can be seen in Table 3.3 and Figure 4.1. Next, the distribution of the MAE error across the different HP configuration setups is shown for each ML algorithm. Subsequently, the results corresponding to S1 that support the answer to the overall research question are presented in Section 4.2.

The results, compared against the CAMS prediction as a baseline, demonstrate that all locally employed algorithms consistently outperform the regional forecast. While the lowest error reduction is observed for the bias-corrected CAMS (CAMS BC), the highest can be noted for the Linear SVR for $PM_{2.5}$ and the GBR for NO_2. Additionally, when local stations were incorporated, a higher error reduction (or explained variability) can be seen, particularly for NO_2 compared against $PM_{2.5}$. Furthermore, a higher performance gain (lower error or higher R^2) can generally be noted when CAMS was included as input. The performance gain is higher for $PM_{2.5}$ than for NO_2. The increase in performance is compared to CAMS is underlined in Section 4.2.1 for $PM_{2.5}$ and Section 4.2.2 for NO_2. By investigating the confusion matrices of the air quality indices, a higher accuracy in predicting pollutant episodes can be additionally observed for the locally applied algorithms. A comparison to Bertrand et al., 2023 shows a higher reduction for NO_2 and similar results for $PM_{2.5}$ for the ML employed in this research. Turning the attention to

S2, the results show that even though the employed ML algorithms continuously yield a lower MAE compared to CAMS while incorporating neighboring stations, most algorithms only show a slight improvement towards their counterpart in S1 for $PM_{2.5}$. Moreover, all algorithms show a higher error than their counterparts for predicting NO_2, while neighboring stations are included.

The subsequent focus is shifted to S3, where historical pollutant measurements at the target station are excluded. The results show that most of the ML algorithms employed to predict a target point from the neighboring stations reduce the overall MAE compared to CAMS, with the highest reduction achieved by the GBR with 15.14% and 18.71% for $PM_{2.5}$ and NO_2, respectively, while incorporating CAMS. If CAMS is not included as additional input, the results are less consistent, though only achieving a consistent performance for $PM_{2.5}$. It should be noted, however, that all locally employed ML algorithms are only optimized on a subset of S1, that might not reflect S2 and S3 as well. Additionally, for other target pollutants, the results could indicate a more positive influence for both scenarios.

In conclusion, it can be seen that while S1 and S2 show a high reduction in comparison to the CAMS prediction, S3 only shows a comparably lower reduction.

Chapter 5 Discussion

In the following, the conclusion of this study will be drawn and discussed, followed by a summary of the work's contributions and an outlook for future work.

5.1. Conclusion

The primary research question addressed in this study was how machine learning (ML) algorithms could improve regional forecasts of the next 23 hours in a local urban environment. The results obtained during Scenario 1 (S1) unequivocally support a positive answer to this question. All locally employed ML algorithms consistently outperformed the regional Copernicus Atmospheric Monitoring Service (CAMS) forecast, with notable improvements in mean absolute error (MAE), root mean squared error (RMSE), and R^2. More particularly, as can be seen in Table 4.4, the Linear support vector regressor (SVR) as best performing locally employed ML algorithms, when compared based on MAE, achieved a MAE of 2.69 $\mu g/m^3$ compared to 3.83 $\mu g/m^3$ by CAMS, representing an error reduction of 29.77% while predicting fine particulate matter with a diameter $< 2.5\mu g/m^3$ ($PM_{2.5}$). A similar picture can be seen for predicting nitrogen dioxide (NO_2). Here, the gradient boosting regressor (GBR) reached a MAE of 5.44 $\mu g/m^3$ compared to 9.89 $\mu g/m^3$ achieved by the regional CAMS forecast, which even displays a reduction of 44.99% towards CAMS, highlighting the particular use of local measurement for predicting NO_2 in urban environments. Even though there are two distinct locally applied ML algorithms achieving the lowest MAE, the margin towards the other ML algorithms is sometimes close for the prediction of NO_2 and nearly negligible for the prediction of $PM_{2.5}$. It is noteworthy that contrary to the implication of the superior prediction performance for the different variations of the artificial neural network (ANN), suggested by a majority of scientific articles throughout the literature, the ANN is not surpassing the prediction performance of the other ML algorithms. Generally, all locally employed ML algorithms benefit from applying model output statistic (MOS) by incorporating the CAMS prediction as additional input, with one exception. The ANN does not seem to be able to make use of the CAMS prediction as additional information to forecast NO_2, achieving the same MAE with and without applying MOS. This might be due to the complex nature of terms of adjustable parameters of the ANN, which were optimized during the hyper parameter (HP) search. Since the search was performed on a subset of three randomly chosen stations, the ANN's HPs might have been overfitted to the specific in-situ measurement stations so that a generalization to other station from the data set was not accomplished as good as for the other ML algorithms. To overcome this adjustment to a selected subset of stations while keeping a similar data set size, one might perform a semi-random selection of individual samples balanced over all stations, seasons, and weekdays to better represent the entirety of the underlying data set.

While interpreting the R^2, a similar but sometimes more distinctive pattern can be seen. The highest R^2 value corresponds to 0.632 or 63.2% of explained variance achieved by the Lasso algorithm compared to 37.9% for the CAMS prediction. This highlights, that the choice of the metric used to measure the performance is essential and might lead to a different best-performing algorithm, even though the underlying prediction for the calculation is the same. For example, the best-performing model for predicting $PM_{2.5}$ with regards to MAE is the GBR whereas the Lasso algorithm outperforms the GBR considering the R^2. The reason for this could be the higher sensitivity of the R^2 to outliers, for example, represented as episodes of pollutant concentrations, and the ability of the Lasso algorithms to capture this episode more accurately. Particularly for NO_2, the R^2 shows a higher improvement towards the CAMS predictions compared to the MAE. The GBR outperforms CAMS with a R^2 of 0.669 compared to -0.011. The negative R^2 of the CAMS prediction shows that the forecast is worse than predicting the mean value of the observed pollutant concentration and indicates that since the CAMS forecast models the regional pollutant concentration, the NO_2 distribution, and dispersion might only affect a proximate area surrounding the pollutant source compared to $PM_{2.5}$, which might exhibit a broader spatial influence. An additional aspect of the air pollution forecast is to capture high pollutant concentrations. Even though the algorithms were not trained to categorize the pollutant concentration by the air quality indices of Table 1.1, it can be seen that the locally employed Linear SVR and GBR outperform the regional CAMS forecast in predicting $PM_{2.5}$ and NO_2, highlighted in Figure 4.6 and Figure 4.11, respectively.

While including neighboring stations to predict the target pollutant concentration at a target station still outperforms the regional CAMS forecast in all employed ML algorithms (see Table 4.5), only some of the algorithms predicting $PM_{2.5}$ can be marginally improved when compared against their counterpart in S1. Moreover, no algorithm can improve the results for incorporating neighboring stations in the prediction of NO_2, even though the same historical data from the target station is also utilized. This underlines the previously made statement that the distribution and dispersion of NO_2 impacts a smaller radius surrounding the pollutant sources than $PM_{2.5}$ and highlights the difficulty in predicting NO_2 in a regional context. Considering the increase in input data (and resulting inference time), including neighboring stations for Scenario 2 (S2) seems not beneficial. However, the fact that the different ML algorithms were only optimized for S1 might influence the results negative. Optimizing the HPs to fit S2 might have a beneficial impact on the results.

Turning attention to the results of Scenario 3 (S3), where the target point is only predicted from the neighboring stations, mixed results can be seen in Table 4.6. While all locally employed algorithms surpass the regional CAMS forecast in predicting $PM_{2.5}$, the NO_2 prediction is only improved (with one exception) if CAMS served as additional input. Still, the locally utilized GBR can exceed the CAMS forecast by 15.14% and 18.71% in predicting $PM_{2.5}$ and NO_2, respectively. Surprisingly, the extreme gradient boosting (XGB), which incorporated all neighboring stations simultaneously, yielded the lowest performance compared to all other ML algorithms, for which the predictions from each neighboring station were only averaged at the target location. Also, including the distance to the target location as an additional feature worsens the results contrary to the expectations (see Table 4.6). It is noteworthy that when no MOS was applied, the performance drop between $PM_{2.5}$ and NO_2 diverges.

While for $PM_{2.5}$, the average pollutant prediction still outperforms the CAMS prediction continuously (e.g., ANN with 10.18%), averaging the prediction for NO_2 does not improve the CAMS prediction. Again, this highlights the local nature of NO_2 compared to $PM_{2.5}$. Investigating other fusion techniques to incorporate the neighboring stations into the prediction at the target location might be beneficial for S2 and particularly S3. While the performance in each of the three scenarios varies, similar distinct patterns for $PM_{2.5}$ and NO_2 can be seen throughout the day (e.g., Figure 4.4 and Figure 4.9). Whereas the prediction MAE for $PM_{2.5}$ gradually increases throughout the day, the NO_2 prediction error shows two peaks in the morning and the evening. The latter is most probably caused by an increased pollutant concentration, as can be seen in Figure 3.4. Considering the hours of the pollutant peak, the high concentration, in turn, is most probably caused by combustion-driven traffic, again suggesting difficulty in estimating on a regional scale. A common pattern in the prediction MAE for both pollutants is the overall increase with rising lead time until the end of the day. This suggests that performing additional predictions throughout the day most probably decreases the prediction error in all scenarios and for all employed ML algorithms by a notable margin. Also, episodes that could not have been expected at midnight could eventually be modeled by the algorithms, making them more adaptable to sudden changes in pollutant concentration.

Focusing on the results obtained during the hyper parameter optimization

(HPO), it has been shown that treating the selection of input features and the number of look back time steps as additional HPs allowed the ML algorithms to match the inputs and the choice of ML dependent HPs, resulting in different subsets of data for each algorithm, which would not have been possible by a preliminary selection of inputs that does not include the different employed ML algorithms. For some of the explored HPs, the algorithm identified parameter values that are close to the border of the search space (e.g., 14.807459 and 14.990312 of the Ridge regressions “alpha” values are close to 15). Here, iteratively, widening the search space might yield a better performance.

Putting the results into the perspective of the compared literature by Bertrand et al., 2023, it is notable that even though the results can be compared with the ones achieved during this research by looking at the percental reduction of each locally applied algorithm towards the regional forecast CAMS, the underlying data set of both studies differs, making the comparison nontrivial. Nevertheless, this research slightly outperformed the locally employed algorithms from Bertrand et al., 2023 regarding $PM_{2.5}$. However, the initial RMSE of 8.6 $\mu g/m^3$ achieved by the CAMS prediction is substantially higher than for the data set utilized in this study with 6.17 $\mu g/m^3$, arguably reducing the margin of improvement in this research. When turning attention to the prediction of NO_2, a more apparent improvement can be noticed for the GBR employed during this research. While Bertrand et al., 2023 achieved an improvement of 33% over the RMSE of the CAMS prediction, the locally employed GBR manages to reduce the error by 45.32%. Arguably, the initial RMSE of the CAMS prediction of NO_2 is with 14.21 $\mu g/m^3$, higher than the one stated by Bertrand et al., 2023 with 12.6% so that the previously made statement for $PM_{2.5}$ now holds in favor of Bertrand et al., 2023 for the prediction of NO_2. Another aspect is the amount of effort spent to collect, merge, and preprocess the underlying data set, which is necessary to answer the research question, particularly concerning MOS, and fulfills the requirements identified during the literature review. Here, a benchmark data set would substantially reduce the time for the research questions to be answered and simplify the comparison across different research from literature.

5.2. Summary of Contributions

It has been shown in this thesis, from different perspectives, that utilizing ML in combination with MOS improves the regional forecast of the target pollutants outstandingly in urban environments, especially when there is a pollutant measurement station at the particular target point. Nevertheless, the results are also promising when the prediction at a target point is interpolated from neighboring stations, showing the potential to yield improved results by incorporating the spatial dimension. In both cases, the regional forecast can be improved, implying the practical use in urban environments to protect people from high pollutant concentrations.

5.3. Future Research

While the application of ML in urban environments to improve regional forecast has been thoroughly investigated from different perspectives, it still offers many aspects that can be explored. Here, the question of how to incorporate the spatial dimension and inter- or extrapolate target points inside an urban environment opens many research opportunities. In this sense, future research could investigate the impact when additional information such as vegetation index, traffic patterns, land use data, or satellite images are provided for the ML algorithms to improve the MOS capability at a particular target point. More data, on the other hand, opens the path to research in more complex models that are capable of capturing the underlying patterns. In particular, an ensemble of algorithms that each capture and represent patterns in different data modalities might yield promising results.

Because one insight suggested in this research is that the target pollutants $PM_{2.5}$ and NO_2 behave differently in terms of dispersion and travel time, analyzing the performance of the proposed ML algorithms for other target pollutants might give additional, target pollutant dependent insights. Furthermore, the publication of a suitable benchmark data set that includes the CAMS forecast and relevant local information is desirable. This would substantially reduce the workload of each subsequent research by providing a directly utilizable data set and simplifying the comparison of different MOS algorithms applied in an urban environment across different research. Another research direction could deal with a feasibility analysis that investigates the performance of the different locally employed algorithms on the fly, unveiling the necessary infrastructure and showing the potential for particular cities when operational.

It has been shown that even though air pollutant prediction has had a long tradition in research, there are still many aspects that might be explored in future research, leaving the research domain of air pollutant prediction an exciting field in literature with many opportunities to identify promising methods that operational applications can adopt.

A Appendix

Attachment

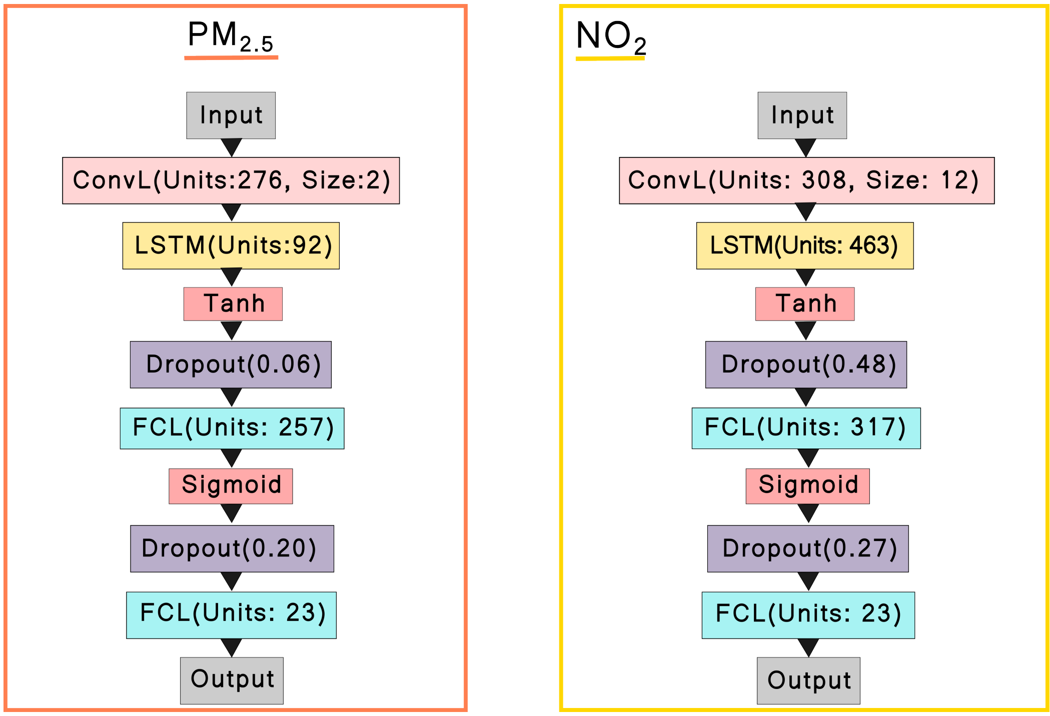


Figure A.1: The neural network architecture found during the HP search for predicting $PM_{2.5}$ (left) and NO_2 (right) are shown. While the number of units per layer sometimes greatly vary, it can be seen that the overall structure of the network follows a similar pattern as when MOS was applied. The higher number of long-short term memory (LSTM) units (463 compared to 92) followed by a stronger regularization through the dropout layer (0.48 compared to 0.06) is noteworthy.

Table A.1: The selected inputs for each of the employed ML algorithms are displayed. It follows similar patterns as Table 4.1 with slight variations (e.g., None of the algorithms predicting NO_2 utilized precipitation).

Input	$PM_{2.5}$					NO_2				
	Lasso	Ridge	SVR	GBR	ANN	Lasso	Ridge	SVR	GBR	ANN
Look back	19	5	2	1	1	9	7	1	1	11
NO_2	✓				✓	✓	✓	✓	✓	✓
NO	✓	✓	✓							
O_3	✓	✓			✓	✓	✓	✓		
$PM_{2.5}$	✓	✓	✓	✓	✓					✓
PM_{10}	✓			✓	✓	✓				✓
SO_2										
Precipitation [mm]	✓	✓	✓	✓	✓					
Temperature [C]	✓	✓	✓	✓	✓	✓	✓		✓	✓
Relative humidity [%]	✓	✓	✓	✓		✓	✓	✓		
Wx	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Wy	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
sine [m]	✓		✓	✓	✓	✓	✓	✓	✓	
cosine [m]				✓	✓		✓	✓	✓	✓
sine [wd]	✓	✓						✓	✓	✓
cosine [wd]	✓	✓	✓	✓		✓	✓		✓	

Table A.2: The results for the HP search, when no MOS was applied are presented. While some HP diverge between $PM_{2.5}$ and NO_2 (the 'tol' for Lasso, Ridge and Linear SVR) others seem to be more consistent (e.g., the loss).

Algorithm	Hyper parameter	Found value	
		$PM_{2.5}$	NO_2
Lasso	alpha	0.193253	0.181867
	tol	0.0008	0.000968
	precompute	True	False
	positive	True	False
	selection	cyclic	random
Ridge	alpha	9.470208	14.651283
	tol	0.000011	0.000805
	solver	auto	sag
Linear SVR	loss	epsilon_insensitive	epsilon_insensitive
	tol	0.000407	0.00002
	C	0.1	1.0
GBR	loss	absolute_error	absolute_error
	learning_rate	0.019516	0.024178
	n_estimators	500	337
	criterion	squared_error	friedman_mse
	min_samples_split	3	6
	min_samples_leaf	7	10
	max_depth	5	9
	max_features	log2	sqrt
	n_iter_no_change	100000	1000
ANN	Batch size	18	12
	Optimizer	RMSprop	RMSprop
	Initial learning rate	0.00001191	0.000581
	Loss	huber_loss	huber_loss
	Learning rate scheduler	True	True
	Start epoch	14	12
	Learning rate decrease	0.810412	0.777385
	Every N epoch	8	4
	Early stopping	True	True
	Patience	12	6

Reference List

Akbal, Y. and Ünlü, K. D. (2022). « A Deep Learning Approach to Model Daily Particular Matter of Ankara: Key Features and Forecasting ». *International Journal of Environmental Science and Technology.*

Aldrin, M and Haff, I (2005). « Generalised Additive Modelling of Air Pollution, Traffic Volume and Meteorology ». *Atmospheric Environment.*

Bergstra, James and Bengio, Yoshua (2012). « Random search for hyper-parameter optimization. » *Journal of machine learning research* 13(2).

Bertrand, Jean-Maxime, Meleux, Frédérik, Ung, Anthony, Descombes, Gaël, and Colette, Augustin (2023). « Improving the European air quality forecast of the Copernicus Atmosphere Monitoring Service using machine learning techniques ». *Atmospheric Chemistry and Physics* 23(9), pp. 5317–5333.

Biancofiore, Fabio, Busilacchio, Marcella, Verdecchia, Marco, Tomassetti, Barbara, Aruffo, Eleonora, Bianco, Sebastiano, Di Tommaso, Sinibaldo, Colangeli, Carlo, Rosatelli, Gianluigi, and Di Carlo, Piero (2017). « Recursive Neural Network Model for Analysis and Forecast of PM10 and PM2.5 ». *Atmospheric Pollution Research.*

Blond, N, Bel, Liliane, and Vautard, Robert (2003). « Three-dimensional ozone data analysis with an air quality model over the Paris area ». *Journal of Geophysical Research: Atmospheres* 108(D23).

Boettger, Carl M. and Smith, Harold J. (1961). « The Nashville daily air pollution forecast ». *Monthly Weather Review.*

Boznar, Marija, Lesjak, Martin, and Mlakar, Primoz (1993). « A Neural Network-Based Method for Short-Term Predictions of Ambient SO2 Concentrations in Highly Polluted Industrial Areas of Complex Terrain ». *Atmospheric Environment. Part B. Urban Atmosphere.*

Brauer, Michael, Brook, Jeffrey R, Christidis, Tanya, Chu, Yen, Crouse, Dan L, Erickson, Anders, Hystad, Perry, Li, Chi, Martin, Randall V, Meng, Jun, et al. (2022). « Mortality–Air Pollution Associations in Low Exposure Environments (MAPLE): Phase 2 ». *Research Reports: Health Effects Institute.*

Breiman, Leo (2001). « Random forests ». *Machine learning* 45, pp. 5–32.

Bundesländer (2023). *Luftmessnetz der Bundesländer.* On Demand. Accessed: 2023-02-21.

Castelli, Mauro, Clemente, Fabiana Martins, Popovič, Aleš, Silva, Sara, and Vanneschi, Leonardo (2020). « A machine learning approach to predict air quality in California ». *Complexity* 2020.

Chang, Yue-Shan, Chiao, Hsin-Ta, Abimannan, Satheesh, Huang, Yo-Ping, Tsai, Yi-Ting, and Lin, Kuan-Ming (2020). « An LSTM-based Aggregated Model for Air Pollution Forecasting ». *Atmospheric Pollution Research.*

Chen, Tianqi and Guestrin, Carlos (2016). « Xgboost: A scalable tree boosting system ». In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining,* pp. 785–794.

Clarke, John F. (1964). « *A* Simple Diffusion Model *for* Calculating Point Concentrations *from* Multiple Sources ». *Journal of the Air Pollution Control Association.*

Comrie, Andrew C. (1997). « Comparing Neural Networks and Regression Models for Ozone Forecasting ». *Journal of the Air & Waste Management Association.*

Copernicus (2023). *Copernicus Atmospheric Monitoring System forecast.* https://ads.atmosphere.copernicus.eu/cdsapp#!/dataset/cams-europe-air-quality-forecasts. Accessed: 2023-03-06.

Cressie, Noel (1993). *Statistics for spatial data.* John Wiley & Sons.

destatis (2023). *Grad der Urbanisierung Deutschland.* https://de.statista.com/statistik/daten/studie/662560/umfrage/urbanisierung-in-deutschland/. Last Accessed: 2023-12-8.

Deutscher Wetterdienst, DWD (2023). *Open Data Bereich des Climate Data Center.* https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/hourly/. Accessed: 2023-02-28.

Di, Qian, Amini, Heresh, Shi, Liuhua, Kloog, Itai, Silvern, Rachel, Kelly, James, Sabath, M. Benjamin, Choirat, Christine, Koutrakis, Petros, Lyapustin, Alexei, Wang, Yujie, Mickley, Loretta J., and Schwartz, Joel (2019). « An Ensemble-Based Model of PM2.5 Concentration across the Contiguous United States with High Spatiotemporal Resolution ». *Environment International.*

Du, Shengdong, Li, Tianrui, Yang, Yan, and Horng, Shi-Jinn (2019). « Deep air quality forecasting using hybrid deep learning framework ». *IEEE Transactions on Knowledge and Data Engineering* 33(6), pp. 2412–2424.

EEA (2023). *European Air Quality Index - Live.* https://airindex.eea.europa.eu/Map/AQI/Viewer/. Last Accessed: 2023-10-30.

Elkamel, A., Abdul-Wahab, S., Bouhamra, W., and Alper, E. (2001). « Measurement and Prediction of Ozone Levels around a Heavily Industrialized Area: A Neural Network Approach ». *Advances in Environmental Research.*

European Environment Agency (2023). *Air pollution: how it affects our health.* URL: https://www.eea.europa.eu/publications/zero-pollution/health/air-pollution (visited on 08/17/2023).

Feng, Xiao, Li, Qi, Zhu, Yajie, Hou, Junxiong, Jin, Lingyan, and Wang, Jingjie (2015). « Artificial Neural Networks Forecasting of PM2.5 Pollution Using Air Mass Trajectory Based Geographic Model and Wavelet Transformation ». *Atmospheric Environment.*

Fisher, Aaron, Rudin, Cynthia, and Dominici, Francesca (2019). « All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. » *J. Mach. Learn. Res.* 20(177), pp. 1–81.

Frenkiel, Franqois N (1956). « Atmospheric Pollution and Zoning in an Urban Area ». *The Scientific Monthly.*

Fronza, G., Spirito, A., and Tonielli, A. (1979). « Real-Time Forecast of Air Pollution Episodes in the Venetian Region. Part 2: The Kalman Predictor ». *Applied Mathematical Modelling.*

Gardner, MW and Dorling, SR (1999). « Neural network modelling and prediction of hourly NOx and NO2 concentrations in urban air in London ». *Atmospheric Environment.*

Glahn, Harry R and Lowry, Dale A (1972). « The use of model output statistics (MOS) in objective weather forecasting ». *Journal of Applied Meteorology and Climatology* 11(8), pp. 1203–1211.

Goodfellow, Ian, Bengio, Yoshua, and Courville, Aaron (2016). *Deep learning.* MIT press.

Harris, Charles R. et al. (2020). « Array programming with NumPy ». *Nature* 585(7825), pp. 357–362.

Hastie, Trevor, Tibshirani, Robert, Friedman, Jerome H, and Friedman, Jerome H (2009). *The elements of statistical learning: data mining, inference, and prediction.* Vol. 2. Springer.

Hinz, Tobias, Navarro-Guerrero, Nicolás, Magg, Sven, and Wermter, Stefan (2018). « Speeding up the hyperparameter optimization of deep convolutional neural networks ». *International Journal of Computational Intelligence and Applications* 17(02), p. 1850008.

Honoré, Cécile, Rouil, Laurence, Vautard, Robert, Beekmann, Matthias, Bessagnet, Bertrand, Dufour, Anne, Elichegaray, Christian, Flaud, Jean-Marie, Malherbe, Laure, Meleux, Frédérik, et al. (2008). « Predictability of European air quality: Assessment of 3 years of operational forecasts and analyses by the PREV'AIR system ». *Journal of Geophysical Research: Atmospheres* 113(D4).

Hu, Xuefei, Belle, Jessica H., Meng, Xia, Wildani, Avani, Waller, Lance A., Strickland, Matthew J., and Liu, Yang (2017). « Estimating PM $_{2.5}$ Concentrations in the Conterminous United States Using the Random Forest Approach ». *Environmental Science & Technology.*

Huang, Chiou-Jye and Kuo, Ping-Huan (2018). « A deep CNN-LSTM model for particulate matter (PM2. 5) forecasting in smart cities ». *Sensors* 18(7), p. 2220.

Hutter, Frank, Hoos, Holger H, and Leyton-Brown, Kevin (2011). « Sequential model-based optimization for general algorithm configuration ». In: *Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers 5.* Springer, pp. 507–523.

Jin, Xue-Bo, Gong, Wen-Tao, Kong, Jian-Lei, Bai, Yu-Ting, and Su, Ting-Li (2022). « A variational Bayesian deep network with data self-screening layer for massive time-series data forecasting ». *Entropy* 24(3), p. 335.

Kim, Ki-Hyun, Kabir, Ehsanul, and Kabir, Shamin (2015). « A review on the human health impact of airborne particulate matter ». *Environment international* 74, pp. 136–143.

Klein, William H and Glahn, Harry R (1974). « Forecasting local weather by means of model output statistics ». *Bulletin of the American Meteorological Society* 55(10), pp. 1217–1227.

Kleine Deters, Jan, Zalakeviciute, Rasa, Gonzalez, Mario, and Rybarczyk, Yves (2017). « Modeling PM $_{2.5}$ Urban Pollution Using Machine Learning and Selected Meteorological Parameters ». *Journal of Electrical and Computer Engineering.*

Li, Tongwen, Shen, Huanfeng, Yuan, Qiangqiang, Zhang, Xuechen, and Zhang, Liangpei (2017). « Estimating Ground-Level PM $_{2.5}$ by Fusing Satellite and Station Observations: A Geo-Intelligent Deep Learning Approach: Deep Learning for PM $_{2.5}$ Estimation ». *Geophysical Research Letters.*

Li, Xiang, Peng, Ling, Yao, Xiaojing, Cui, Shaolong, Hu, Yuan, You, Chengzeng, and Chi, Tianhe (2017). « Long Short-Term Memory Neural Network for Air Pollutant Concentration Predictions: Method Development and Evaluation ». *Environmental Pollution.*

Liang, Yuxuan, Ke, Songyu, Zhang, Junbo, Yi, Xiuwen, and Zheng, Yu (2018). « Geoman: Multi-level attention networks for geo-sensory time series prediction. » In: *IJCAI*. Vol. 2018, pp. 3428–3434.

Lindauer, Marius, Eggensperger, Katharina, Feurer, Matthias, Biedenkapp, André, Deng, Difan, Benjamins, Carolin, Ruhkopf, Tim, Sass, René, and Hutter, Frank (2022). « SMAC3: A Versatile Bayesian Optimization Package for Hyperparameter Optimization. » *J. Mach. Learn. Res.* 23(54), pp. 1–9.

Mao, Wenjing, Wang, Weilin, Jiao, Limin, Zhao, Suli, and Liu, Anbao (2021). « Modeling air quality prediction using a deep learning approach: Method optimization and evaluation ». *Sustainable Cities and Society* 65, p. 102567.

Marécal, V. et al. (2015). « A Regional Air Quality Forecasting System over Europe: The MACC-II Daily Ensemble Production ». *Geoscientific Model Development.*

Martín Abadi et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.* Software available from tensorflow.org.

McCollister, George M. and Wilson, Kent R. (1975). « Linear Stochastic Models for Forecasting Daily Maxima and Hourly Concentrations of Air Pollutants ». *Atmospheric Environment (1967).*

McKinney, Wes (2010). « Data Structures for Statistical Computing in Python ». In: *Proceedings of the 9th Python in Science Conference.* Ed. by Stéfan van der Walt and Jarrod Millman, pp. 56–61.

Meng, Xia, Liu, Cong, Zhang, Lina, Wang, Weidong, Stowell, Jennifer, Kan, Haidong, and Liu, Yang (2021). « Estimating PM2. 5 concentrations in Northeastern China with full spatiotemporal coverage, 2005–2016 ». *Remote sensing of environment* 253, p. 112203.

Molnar, Christoph (2020). *Interpretable machine learning.* Lulu. com.

Muthukumar, Pratyush, Cocom, Emmanuel, Nagrecha, Kabir, Comer, Dawn, Burga, Irene, Taub, Jeremy, Calvert, Chisato Fukuda, Holm, Jeanne, and Pourhomayoun, Mohammad (2021). « Predicting PM2. 5 atmospheric air pollution using deep learning with meteorological data and ground-based observations and remote-sensing satellite big data ». *Air Quality, Atmosphere & Health,* pp. 1–14.

Pedregosa, F. et al. (2011). « Scikit-learn: Machine Learning in Python ». *Journal of Machine Learning Research* 12, pp. 2825–2830.

Prakash, Amit, Kumar, Ujjwal, Kumar, Krishan, and Jain, V. K. (2011). « A Wavelet-based Neural Network Model to Predict Ambient Air Pollutants' Concentration ». *Environmental Modeling & Assessment.*

Qiao, Weibiao, Tian, Wencai, Tian, Yu, Yang, Quan, Wang, Yining, and Zhang, Jianzhuang (2019). « The forecasting of PM2. 5 using a hybrid model based on wavelet transform and an improved deep learning algorithm ». *IEEE Access* 7, pp. 142814–142825.

Qin, Dongming, Yu, Jian, Zou, Guojian, Yong, Ruihan, Zhao, Qin, and Zhang, Bo (2019). « A novel combined prediction scheme based on CNN and LSTM for urban PM 2.5 concentration ». *IEEE Access* 7, pp. 20050–20059.

Russell, Stuart J (2010). *Artificial intelligence a modern approach.* Pearson Education, Inc.

Saez, Marc and Barceló, Maria A (2022). « Spatial prediction of air pollution levels using a hierarchical Bayesian spatiotemporal model in Catalonia, Spain ». *Environmental Modelling & Software* 151, p. 105369.

Tao, Qing, Liu, Fang, Li, Yong, and Sidorov, Denis (2019). « Air pollution forecasting using a deep learning model based on 1D convnets and bidirectional GRU ». *IEEE access* 7, pp. 76690–76698.

Navarro-Guerrero, Nicolás (2014). *Yet Another Thesis Template.* URL: https://bitbucket.org/nicolas-navarro-guerrero/thesis-template.

Tian, Jiawei, Liu, Yan, Zheng, Wenfeng, and Yin, Lirong (2022). « Smog prediction based on the deep belief-BP neural network model (DBN-BP) ». *Urban Climate* 41, p. 101078.

Umweltbundesamt, UBA (2023a). *Luftmessnetz des Bundes.* On Demand. Accessed: 2023-02-21.

Umweltbundesamt, UBA (2023b). *Metadaten des Bundesluftmessnetzes.* Online. Accessed: 2023-03-20.

Whitaker (2023). https://github.com/Unidata/netcdf4-python.

Wilczak, J, McKeen, S, Djalalova, I, Grell, G, Peckham, S, Gong, W, Bouchet, V, Moffet, R, McHenry, J, McQueen, J, et al. (2006). « Bias-corrected ensemble and probabilistic forecasts of surface ozone over eastern North America during the summer of 2004 ». *Journal of Geophysical Research: Atmospheres* 111(D23).

World Health Organization et al. (2021). *WHO global air quality guidelines: particulate matter (PM2. 5 and PM10), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide.* World Health Organization.

World Health Organization (2022). *Air Pollution: Overview.* URL: https://www.who.int/health-topics/air-pollution#tab=tab_1 (visited on 08/25/2022).

Xarray developers (2023). https://github.com/pydata/xarray.

Yildirim, Yilmaz and Bayramoglu, Mahmut (2006). « Adaptive Neuro-Fuzzy Based Modelling for Prediction of Air Pollution Daily Levels in City of Zonguldak ». *Chemosphere.*

Zamani, Mehdi, Cao, Chunxiang, Ni, Xiliang, Bashir, Barjeece, and Talebiesfandarani, Somayeh (2019). « PM2. 5 prediction based on random forest, XGBoost, and deep learning using multisource remote sensing data ». *Atmosphere* 10(7), p. 373.

Zeng, Yongkang, Chen, Jingjing, Jin, Ning, Jin, Xiaoping, and Du, Yang (2022). « Air quality forecasting with hybrid LSTM and extended stationary wavelet transform ». *Building and Environment.*

Zhang, Luo, Liu, Peng, Zhao, Lei, Wang, Guizhou, Zhang, Wangfeng, and Liu, Jianbo (2021). « Air Quality Predictions with a Semi-Supervised Bidirectional LSTM Neural Network ». *Atmospheric Pollution Research.*

Zhao, Jiachen, Deng, Fang, Cai, Yeyun, and Chen, Jie (2019). « Long short-term memory-Fully connected (LSTM-FC) neural network for PM2. 5 concentration prediction ». *Chemosphere* 220, pp. 486–492.

Zheng, Yu, Yi, Xiuwen, Li, Ming, Li, Ruiyuan, Shan, Zhangqing, Chang, Eric, and Li, Tianrui (2015). « Forecasting Fine-Grained Air Quality Based on Big Data ». In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM.

Zhou, Yanlai, Chang, Fi-John, Chang, Li-Chiu, Kao, I-Feng, and Wang, Yi-Shin (2019). « Explore a deep learning multi-output neural network for regional multi-step-ahead air quality forecasts ». *Journal of cleaner production* 209, pp. 134–145.

B Appendix

Acknowledgements

I would like to thank my supervisors Prof. Dr. Dr. h.c. Frank Kirchner and Dr. Nicolás Navarro-Guerrero, who made it possible to conduct my research by providing me the freedom to choose the research topic for this thesis. Additional thanks go to Prof. Dr. Dr. h.c. Frank Kirchner for providing the infrastructure without which the various calculations for this thesis would have not been possible in time. Special thanks also go to Dr. Nicolás Navarro-Guerrero for the fruitful discussion that helped to guide my thesis in the right directions. Finally, I would like to thank all the people that supported me in their own ways.

Glossary

Symbols

CO

carbon monoxide. 1

NO

nitrogen monoxide. 41

NO_2

nitrogen dioxide. 1, 4, 12, 22, 23, 25, 26, 35–38, 41, 42, 44–47, 51–56, 58, 59, 62–73

O_3

ozone. 1, 4, 6, 13, 22, 23

PM

particulate matter. 1, 3, 4, 7, 22, 35

PM_{10}

particulate matter with a diameter $< 10\mu g/m^3$. 1, 4, 12, 41

$PM_{2.5}$

fine particulate matter with a diameter $< 2.5\mu g/m^3$. 1, 2, 4, 5, 9, 11–16, 18, 21, 22, 25, 26, 28, 35–38, 41, 42, 44–51, 53, 56, 58–60, 62, 64–71, 73

SO_2

sulfur dioxide. 1, 3, 4, 16, 22, 23, 41

A

ANN

artificial neural network. 3, 4, 13, 15, 20, 29–32, 34, 38, 41–45, 47–54, 56–58, 60–64, 66–68

AOD

aerosol optical depth. 9–11, 15, 16, 18

C

CAMS

Copernicus Atmospheric Monitoring Service. 8, 21–23, 28, 29, 35–38, 40, 42, 45–70

CGM

Convolutional Generalization Model. 13

CNN

convolutional neural network. 11, 14–16, 30, 34, 35, 38

CTM

chemical transport model. 1, 2, 4, 5, 8, 9

D

DBN

deep belief network. 14, 15

DWD

Deutscher Wetterdienst. 21, 22, 28, 38

E

EEA

European Environmental Agency. 2, 5, 51, 55

G

GBR

gradient boosting regressor. 12, 15, 30, 38, 41–43, 45, 47, 49, 52–55, 58–69

GLMM

generalized linear mixed model. 15

GRF

Gaussian Random Field. 15

GRU

Gated Recurrent Unit. 14

H

HP

hyper parameter. 5, 20, 31–35, 39–43, 45, 56, 58, 64, 67–69, 71, 73

HPO

hyper parameter optimization. 29, 31, 32, 34, 38, 40, 42, 44, 45, 64, 68

L

LSTM

long-short term memory. 13–16, 19, 30, 32, 34, 38, 71

M

MAE

mean absolute error. 4, 5, 16, 19, 22, 37–39, 42, 45–68

MF

meteorological fields. 9

ML

machine learning. 2, 5–7, 12, 16, 19–21, 28–32, 35–43, 45–55, 58–61, 63–70, 72

MOS

model output statistic. 2–6, 12, 30, 36–40, 47–50, 53, 54, 57, 59–64, 66–71, 73

MSD

mean signed deviation. 38, 39, 48–50, 52–54, 57, 60–64

N

NDVI

normalized difference vegetation index. 9

R

RBM

restricted Boltzmann machine. 14

RF

random forest. 10, 12, 15, 16, 32

RMSE

root mean squared error. 4, 16, 18, 19, 38, 45–47, 66, 69

S

S1

Scenario 1. 35–38, 40, 45, 46, 55–57, 60–68

S2

Scenario 2. 35, 37–40, 55–58, 63, 65, 68

S3

Scenario 3. 35, 37, 39, 40, 58–60, 62, 63, 65, 68

SAE

stacked auto-encoder. 14

SMAC

sequential model-based algorithm configuration. 32, 38

SVM

support vector machine. 12, 19

SVR

support vector regressor. 16, 30, 38, 42, 43, 49–51, 60, 64, 66, 67, 73

U

UBA

Umweltbundesamt. 21, 22, 28, 38

USA

united states of America. 3, 6

W

WHO

World Health Organization. 1, 2

X

XGB

extreme gradient boosting. 15, 30, 37, 58, 59, 68