

## Master Thesis

# Improving the Availability of Contextual Data with Machine Learning-Based Interpolation

**Ian Maurice Buck**

---

ian.buck@studium.uni-hamburg.de  
Study Program Information Systems  
Matr.-Nr. 6911467

First Reviewer: Prof. Dr. Janick Edinger  
Second Reviewer: Philipp Kisters

Submission Deadline: 03.09.2023

The question of whether a computer can think is no more  
interesting than the question of whether a submarine can swim.  
– *Edsger Dijkstra*

# Abstract

In 2023 56% of the human population already lives in urban areas with the number projected to continuously increase to 68% by 2050. Combined with the ongoing climate change and urban densification due to the need for more and more living space, cities are facing many new challenges. With the removal of vegetation in favour of living space and the sealing of surfaces with heat-absorbing materials such as asphalt or concrete for streets and highways, rising temperatures lead to new phenomena that pose risks for the urban citizens. One especially critical phenomenon is the Urban Heat Island (UHI).

To detect and understand UHI formation, one must measure the urban climate of a city in a very detailed way, which current official meteorological monitoring networks with less than one weather station per city are not capable of. In this work, we explore how machine-learning based interpolation can be used in urban air temperature ( $T_{air}$ ) sensing applications to leverage new possibilities of private-owned weather station and sensor networks to interpolate  $T_{air}$  either for specific locations, augmenting times when sensors might be offline, or non-stationary sensors moving through a city, as well as areal interpolation to predict  $T_{air}$  between sensors. We show that for interpolating a single sensor, Histogram-based Gradient Boosting (HistGB) is a powerful machine learning approach that achieves a root mean squared error (RMSE) between 0.4 and 0.5 for sensors from Neatmo and SensorCommunity in Hamburg and Stuttgart, and that other features other than  $T_{air}$  of surrounding neighbours have little to no influence on the model. Areal interpolation on the other hand seems more challenging and performs slightly worse with an RMSE of 1.4-1.5 during especially hot times during the day with R-squared ( $R^2$ ) values at most of 0.2, while during the night the RMSE is down to 0.65-0.7 for HistGB with  $R^2$  scores around 0.7, suggesting that there is more improvement potential in this area.



# Table of Contents

<b>List of Figures</b>	<b>v</b>
<b>List of Tables</b>	<b>vii</b>
<b>List of Abbreviations</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Objective . . . . .	2
1.2 Structure of this work . . . . .	3
<b>2 Related Work</b>	<b>5</b>
2.1 Urban Heat Islands . . . . .	5
2.1.1 UHI Classification . . . . .	5
2.1.2 Surface Urban Heat Island . . . . .	7
2.1.3 Canopy Urban Heat Island . . . . .	8
2.2 Smart Cities . . . . .	9
2.2.1 Sensing Layer . . . . .	10
2.2.2 Application Layer . . . . .	12
2.3 Interpolation . . . . .	13
2.3.1 Exact Point Interpolation . . . . .	14
<b>3 Machine Learning-based Interpolation</b>	<b>17</b>
3.1 Machine-Learning Application Areas in Air Temperature Interpolation . . . . .	18
3.2 Model Selection Criteria . . . . .	19
3.3 Comparison of Machine Learning Algorithms . . . . .	22
3.3.1 (Multi) Linear Regression . . . . .	22
3.3.2 KNN Regression . . . . .	23
3.3.3 Regression Trees and Random Forests . . . . .	23
3.3.4 Histogram-Based Gradient Boosting . . . . .	26
3.3.5 Support Vector Machine Regression . . . . .	26
3.3.6 Neural Networks . . . . .	27
<b>4 Preparation of Datasets</b>	<b>29</b>
4.1 Private Weather Station Network Providers . . . . .	30
4.1.1 Sensor.Community . . . . .	30
4.1.2 Netatmo . . . . .	32

---

4.1.3	Other Providers . . . . .	35
4.2	Reference Data Providers . . . . .	35
4.2.1	DWD . . . . .	36
4.2.2	Locally Operated Weather Stations . . . . .	37
4.3	Remote Sensing Data Providers . . . . .	39
4.3.1	Google Earth Engine . . . . .	40
4.4	Quality Control . . . . .	40
4.4.1	Quality Control for Sensor.Community . . . . .	41
4.4.2	Quality Control for Netatmo . . . . .	45
4.5	Feature Engineering . . . . .	45
4.5.1	Feature Engineering Pipelines and Automation . . . . .	46
4.5.2	Feature Overview . . . . .	47
4.5.3	Features used in this Work . . . . .	50
4.6	Additional Considerations . . . . .	52
<b>5</b>	<b>Evaluation</b>	<b>53</b>
5.1	Interpolation of Air Temperature for a Specific Location . . . . .	54
5.1.1	Model Comparison . . . . .	55
5.1.2	Further Evaluation - Gradient Boosting . . . . .	56
5.2	Areal Interpolation of Air Temperature . . . . .	62
5.2.1	Model Comparison . . . . .	64
5.2.2	Comparison with Kriging . . . . .	64
5.2.3	Comparison during Night . . . . .	66
5.2.4	Comparison with Netatmo . . . . .	68
<b>6</b>	<b>Conclusion</b>	<b>71</b>
6.1	Future Outlook . . . . .	72
<b>Bibliography</b>		<b>xi</b>
<b>Appendix</b>		<b>xxiii</b>
1	Netatmo . . . . .	xxiii
2	Sklearn Model Parameters for Single Station Interpolation . . . . .	xxiv
3	Feature Engineering . . . . .	xxiv
4	Histogram-based Gradient Boosting Single Location Interpolation . . . . .	xxiv
5	Areal Interpolation . . . . .	xxv
6	Sensor.Community . . . . .	xxx
<b>Eidesstattliche Versicherung</b>		<b>xxxiii</b>

# List of Figures

2.1	Mesoscale view of the urban climate, adopted from [Oke06] . . . . .	6
2.2	Local scale view of the urban climate, adopted from [Oke06] . . . . .	6
2.3	Microscale view of the urban climate, adopted from [Oke06] . . . . .	7
2.4	In the data layer (left), a wide variety of environmental data is collected with the help of multiple sensors. These are connected to their citizen-owned local base stations, which manage access rights and forward collected data to subscribed services (right) via the decentralized publish-subscribe in the network layer (centre). . . . .	10
4.1	Temperature map from Sensor Community for Hamburg, Germany, on 22.06.2023 12:51h with the German Weather Service (DWD) reference at 25°C . . . . .	31
4.2	Temperature outlier from Sensor Community for Hamburg, Germany, on 22.06.2023 12:51h with the DWD reference at 25°C . . . . .	31
4.3	Sensor locations of Sensor Community in Germany, as of 01.05.2023, of sensor type DHT22 (2590 sensors), BME280 (1558 sensors), BMP280 (100 sensors), BMP180 (72 sensors) . . . . .	32
4.4	Sensor locations of Netatmo in Hamburg, Germany, as of 28.06.2023 . . . . .	34
4.5	Sensor locations of Netatmo in Stuttgart, Germany, as of 19.06.2023 . . . . .	34
4.6	Temperature sensor locations from Weather Observations Website (WOW), accessed on 05.07.2023 . . . . .	35
4.7	DWD Weather Station Locations in Germany, <a href="https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/subdaily/standard_format/KL_Standardformate_Beschreibung_Stationen.txt">https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/subdaily/standard_format/KL_Standardformate_Beschreibung_Stationen.txt</a> , accessed 28.06.2023 . . . . .	37
4.8	Weather Station Locations in Stuttgart, <a href="https://www.stadtklima-stuttgart.de/">https://www.stadtklima-stuttgart.de/</a> , last accessed: 10.08.2023 . . . . .	38
4.9	QC Results for Sensor.Community Data for Germany, January 2023 . . . . .	42
4.10	QC Results for Sensor.Community Data for Germany, June 2023 . . . . .	42
4.11	QC Results for Sensor.Community for Germany, June 2023 . . . . .	43
4.12	Quality Control (QC) Result Statistics for Netatmo Data for Hamburg, June 2023 . . . . .	45
5.1	RMSE by Model Type with the Confidence Interval of 95% . . . . .	55

5.2	RMSE for Increasing Minimum Distance with 10 Neighbours, Hamburg, Netatmo . . . . .	57
5.3	RMSE for Increasing Minimum Distance with 10 Neighbours By Station Id, Hamburg, Netatmo . . . . .	57
5.4	RMSE based on Features Selected with 30 Neighbours, Hamburg, Netatmo	59
5.5	Permutation Importance for Single Station on 5-Fold Cross Validation, Hamburg, Netatmo . . . . .	59
5.6	QC Level Comparison for Netatmo Data for Hamburg, June 2023 . . . . .	60
5.7	Comparison between amount of data sampled from simulated moving sensors for Netatmo station 70ee:50:83:b1:e2 with 30 neighbours . . . . .	61
5.8	Areal Interpolation Comparison, Sensor.Community Stuttgart, 19.06.2023 2PM . . . . .	63
5.9	Locations for Sensor.Community around Stuttgart, Germany, June 2023 . . . . .	63
5.10	Ordinary Kriging Spherical Variogram, Sensor.Community Stuttgart, 19.06.2023 2PM . . . . .	65
5.11	HistGradientBoostingRegressor Areal Interpolation, Sensor.Community Stuttgart, 19.06.2023 2PM . . . . .	65
5.12	Areal Interpolation Comparison, Sensor.Community Stuttgart, 19.06.2023 5AM . . . . .	67
5.13	HistGradientBoostingRegressor Areal Interpolation, Sensor.Community Stuttgart, 19.06.2023 5AM . . . . .	67
5.14	Ordinary Kriging Power Variogram, Netatmo Stuttgart, 19.06.2023 2PM . . . . .	68
1	QC for Netatmo in Hamburg showing difference between day and night, 19.06.2023 . . . . .	xxiii
2	Evaluation Single Station Interpolation, Detailed View . . . . .	xxvi
3	Evaluation Single Station Interpolation, Detailed View . . . . .	xxvii
4	Netatmo Stations for Minimum Distance Between Stations, Hamburg . . . . .	xxviii
5	DWD Station 4931, Stuttgart, June 2023 . . . . .	xxix
6	DWD Station 4931, Stuttgart, 19. June 2023 . . . . .	xxix

# List of Tables

4.1	Netatmo Sensor Specifications (Vendor reported) . . . . .	33
4.2	Quality Control Steps of CrowdQC+ . . . . .	44
4.3	ICDC Dataset Parameters . . . . .	48
4.4	Indexes used by Alonso and Renard [AR20] to predict $T_{air}$ . . . . .	49
4.5	Features for Air Temperature Interpolation used by Venter et al. [VBEM20]	49
4.6	Features for Air Temperature Interpolation Used in this Work . . . . .	51
5.1	RMSE and $R^2$ of Ordinary Kriging for Stuttgart 19.06.2023 2PM, Sensor.Community	66
5.2	RMSE and $R^2$ of machine learning (ML) Models for Stuttgart 19.06.2023 2PM, Sensor.Community . . . . .	66
5.3	RMSE and $R^2$ of Ordinary Kriging for Stuttgart 19.06.2023 2PM, Netatmo	69



## List of Abbreviations

AI	artificial intelligence
ANN	Artificial Neural Network
BUHI	Boundary Layer Urban Heat Island
CUHI	Canopy Urban Heat Island
DL	deep learning
DWD	German Weather Service
ECV	Essential Climate Variables
EUMETNET	European Meteorological Network
Google EE	Google Earth Engine
HistGB	Histogram-based Gradient Boosting
IDW	inverse-distance weighting
KNN	K-Nearest Neighbours
LCS	low-cost sensors
LCZ	local climate zone
LST	Land Surface Temperatures
ML	machine learning
MODIS	Moderate Resolution Imaging Spectroradiometer
MSE	mean squared error
NaN	not a number

NDVI	.....	Difference Vegetation Index
OK	.....	Ordinary Kriging
P2P	.....	peer-to-peer
PBL	.....	planetary boundary layer
PM	.....	particulate matter
PWS	.....	private weather station
QC	.....	Quality Control
QRF	.....	Quantile Regression Forests
R <sup>2</sup>	.....	R-squared
RBF	.....	Radial Basis Function
RF	.....	Random Forests
RFSI	.....	Random Forest for Spatial Interpolation
RFsp	.....	Random Forest for Spatial Predictions Framework
RMSE	.....	root mean squared error
SANE	.....	Smart Networks for Urban Citizen Participation
SUHI	.....	Surface Urban Heat Islands
SVF	.....	sky-view factor
SVM	.....	Support Vector Machine
SVR	.....	Support Vector Regression
T <sub>air</sub>	.....	air temperature
UHI	.....	Urban Heat Island
UHII	.....	urban heat island intensity
UX	.....	user experience
VIF	.....	variance inflation factor
WMO	.....	World Meteorological Organization
WOW	.....	Weather Observations Website
WSN	.....	Wireless sensor networks

---

# 1 Introduction

In 2023 56% of the human population already lives in urban areas with the number projected to continuously increase to 68% by 2050 [UNSA19]. Combined with the ongoing climate change and urban densification due to the need for more and more living space, cities are facing many new challenges. With the removal of vegetation in favour of living space and the sealing of surfaces with heat-absorbing materials such as asphalt or concrete for streets and highways [GRGTDW20], rising temperatures lead to new phenomena that pose risks for the urban citizens. One especially critical phenomenon is the UHI. A UHI is a local occurrence where temperatures are higher than in surrounding rural areas, posing health risks, especially for the elderly, children, or citizen with prior health-issues [MBG15], negatively impacting pedestrians' comfort and other city-related topics such as water- and energy-management. The research topic of UHI's has seen a huge number of contributions in the last two decades, but according to Steward, *controlled measurement* and *openness of method* are still two mayor areas of weakness [Ste11], that are related to difficulties in taking measurements in urban areas [Oke06] and a lack of rigorous methodology.

There are two mayor approaches to measure the temperature of a city. The first approach is to use satellites to measure Land Surface Temperatures (LST) [PPC<sup>+</sup>12]. While allowing for an analysis of large areas without the need of ground weather-stations, this approach comes with certain downsides, such as low temporal and spatial resolution and restrictions such as only being able to measure temperatures when no clouds interfere with the microwaves send from the measuring satellite [ZPL15]. The exact spatial and temporal resolution depends on the type of satellite used, with spatial resolutions of older satellites such as Moderate Resolution Imaging Spectroradiometer (MODIS) ranging from  $1\text{km}^2$  to  $5\text{km}^2$ , while newer satellites such as LANDSAT or Sentinel 2 offer higher spatial resolutions between  $10\text{m}^2$  to  $50\text{m}^2$  per pixel. In all cases, temporal resolutions range from daily to monthly temperature values [GVP], depending how often the satellites pass over a certain area. These temporal resolutions are not enough to capture the microclimate of a city [VS17].

In comparison, traditional meteorological observation networks, such as operated by the DWD, offer a much higher temporal resolution at 10 min intervals at 2m and 5cm using ground weather stations; however, they are usually only available at very low spatial resolutions, as they are used to monitor the climate at a meso-scale level. Additionally, the placement of these stations is usually not optimized for the detection of UHI's, as they are commonly placed near to airports that are not located directly in the city centre.

---

They can however be used as reference stations to get an idea about the boundaries of the climate inside a city as they offer high-quality data by using high-quality reference sensors and follow World Meteorological Organization (WMO) guidelines [Oke06].

Lastly, there is the possibility of deploying sensor networks to closely monitor the climate of the city. These sensor networks can either be deployed professionally by the city itself or research projects for a limited period, in that case called testbeds, or they can be deployed by citizens themselves, in that case called citizen-owned sensor networks or private weather station (PWS) networks in the case of crowd-sensing meteorological data. Well-known examples of professionally setup testbeds include the Birmingham Urban Climate Laboratory (BUCL) [CMY<sup>+</sup>15] and the Helsinki Testbed [KPS<sup>+</sup>11] that usually focus on measuring meso-scale weather phenomena and are very costly to run and maintain. PWS networks can either be run by citizens themselves, such as the Sensor.Community<sup>1</sup> project, or by companies, such as Netatmo<sup>2</sup>; however, citizens are usually directly responsible for the placement and maintenance of the individual sensors. Due to the lack of quality control, the data quality of these networks is usually not as high as the data quality of professional networks and require special data QC steps [FBD<sup>+</sup>21, MFG<sup>+</sup>17]; however, they offer a high spatial resolution depending on the provider and can be used to gain insights into the microclimate of a city. Recently, there have been efforts to combine the data from PWS networks, mainly from Netatmo, WeatherUnderground, and Weather Underground, with data from national weather services, such as the DWD, to improve weather prediction quality. The main collaboration network in this area is European Meteorological Network (EUMETNET) which includes 31 European national meteorological services [HGMS<sup>+</sup>22].

While these different approaches offer different advantages and disadvantages of measuring  $T_{air}$  on the ground, they all have one thing in common: they only offer point measurements of the temperature at the location of the sensor. To get an overview of the temperature distribution across a city, various interpolation methods are needed to for example create a continuous data-layer from single point measurements or interpolate missing data for individual sensors.

## 1.1 Objective

The main objective for this work is to explore the feasibility of the usage of ML models for  $T_{air}$  interpolation in local urban environments. As part of this exploration, two main use cases are discussed, namely  $T_{air}$  interpolation for a single station, and areal interpolation for a wider urban environment. The main idea of  $T_{air}$  interpolation for a single location is to train a ML model for that specific location and capture the relationship to surrounding neighbour sensors, which can then be used to either impute missing values for a sensor

---

<sup>1</sup><https://deutschland.maps.sensor.community/>, last accessed: 22.08.2023

<sup>2</sup><https://weathermap.netatmo.com/>, last accessed: 22.08.2023

if that sensor is offline, or if there is no stationary sensor for that specific location to begin with, e.g., a non-stationary sensor that moves through the urban city, to impute values while no moving sensor is currently in the area. Especially the moving sensor case could be interesting, as this could be a solution to improving the limited spatial coverage of a sensor network.

Next, areal interpolation of  $T_{air}$  is important as many research related activities commonly rely on continuous or gridded data fields in order to do analysis, and interpolation is a way of turning sensor readings at discrete locations into a gridded  $T_{air}$  map. The main challenge of this approach is that there are no sensors in every location, making it hard to train supervised ML models and validate interpolation results. For this approach, commonly collected weather information, such as temperature, humidity, rain, pressure, and wind, are used in conjunction with remote sensing features such as vegetation indexes [AR20], that can indicate similarities between the environments in which the individual sensors are placed.

Next to discussing the individual technical capabilities of ML models, data plays an important role of when training, testing, and operating ML models. To collect urban weather data, different PWS providers are compared, and data collected from Netatmo and SensorCommunity. Data pre-processing steps as well as QC is discussed to guarantee good data quality and reliable evaluation results. Additionally, feature engineering steps are discussed to capture additional information such as location, time, and more.

After exploring available ML models and collecting data for training and testing, the last goal is to do an evaluation of the different ML models for both use cases to determine the feasibility and identify prediction quality and rank model performances by assessing root mean squared error (RMSE) and  $R^2$  scores.

## 1.2 Structure of this work

The rest of the thesis is structured as follows: Chapter 2 begins with an introduction on related work. The focus topics are UHIs, one of the main motivating factors behind this work, Smart City and Sensor Networks, in order to identity new capabilities in a smart and connected urban sphere, and interpolation techniques from statistics and geo-statistics. In Chapter 3, ML-based interpolation is introduced with a discussion of model selection criteria for  $T_{air}$  interpolation and a comparison between different ML regression models which can be used for interpolation. Chapter 4 discusses data provider and collection, data pre-processing steps and quality control, as well as some feature engineering aspects. Lastly, the evaluation is done in Chapter 5, where different ML models are implemented and trained and important questions are discussed, such as among others feasibility, model performances, and feature importance. Finally, Chapter 6 discusses the findings of this thesis and gives an outlook into future work and research directions.

---



## 2 Related Work

Before we can start to investigate the usage of machine-learning based interpolation techniques in the context of urban climate data, we first need to understand what UHIs are and how they can be classified and detected to define requirements for the ML models later. Especially in the context of Smart Cities with new possibilities such as sensor networks, we need to understand how urban climate data can be collected and what challenges arise such as spatial and temporal data availability, data quality and more. Due to the complexity of the urban climate [Oke06], special domain knowledge is needed to understand the data and the underlying processes. Finally, we need to get an understanding of existing interpolation techniques, traditionally in the form of regression analyses or in the context of climate data, geostatistical models, to define a baseline for the evaluation of ML-based models.

### 2.1 Urban Heat Islands

UHIs have been the centre of a lot of attention for quite some time in the scientific community. As early as 1833, with the research of Luke Howard in London who observed higher temperatures inside London than in surrounding areas [How33], UHIs have seen a steadily increase in scientific contributions. The term *Urban Heat Island* was first introduced in the 1940s [BP47]. The recording and investigation of UHIs has seen major steps since the begin of modern climatology, also known as the Sundborg's era beginning with Sundborg's 1951 classic heat island study of Uppsala [Sun51]. UHIs occur in many cities around the globe [PPC<sup>+</sup>12] in different climatic zones, during different times of day and in different intensities.

UHIs are so important, because heat related deaths are rising across the globe [KH08] and extreme heat waves are projected to occur more often and extreme with the ongoing climate change [LSF19]. Heat also has significant impact on human performance [KBF<sup>+</sup>16], mental health [OMPR18], or can disrupt sleep [OMMF17] and causes other issues such as overheating that causes urban infrastructure to fail, decreased air quality and low outdoor thermal comfort levels [SJVL<sup>+</sup>13].

#### 2.1.1 UHI Classification

UHIs can be classified in many ways. Typically, there is a horizontal classification, defining the superficial extension of the UHI from micro-, to local- to meso-scale, and a ver-

---

tical classification, defining in which vertical layer of the urban area the heat island is observed. To better understand these scales and the anatomy of the planetary/urban boundary layer, Figures 2.1, 2.2, and 2.3 show a detailed view of the meso-, local- and micro-scale of the urban climate respectively, as illustrated by Oke 2006 [Oke06].

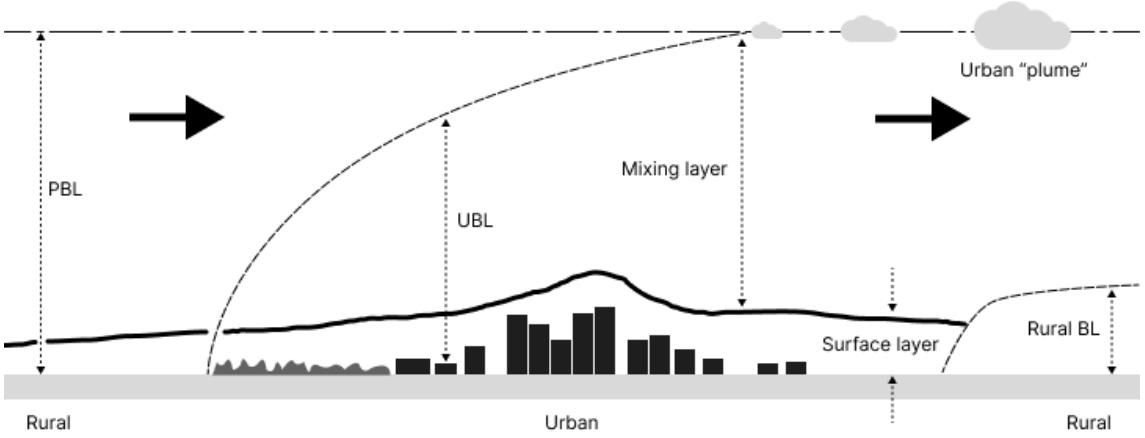


Figure 2.1: Mesoscale view of the urban climate, adopted from [Oke06]

The mesoscale, as depicted in Fig 2.1, spans the whole urban environment of a city, typically tens of kilometres. There are several boundary layers, that comprise different scales. The planetary boundary layer (PBL) [Wyn85] is the lowest layer of the Earth's atmosphere and spans from the surface to a height of several hundred meters up to several kilometres. It is characterised by the turbulent mixing of air, forming wind currents, that are mainly influenced by the underlying surface.

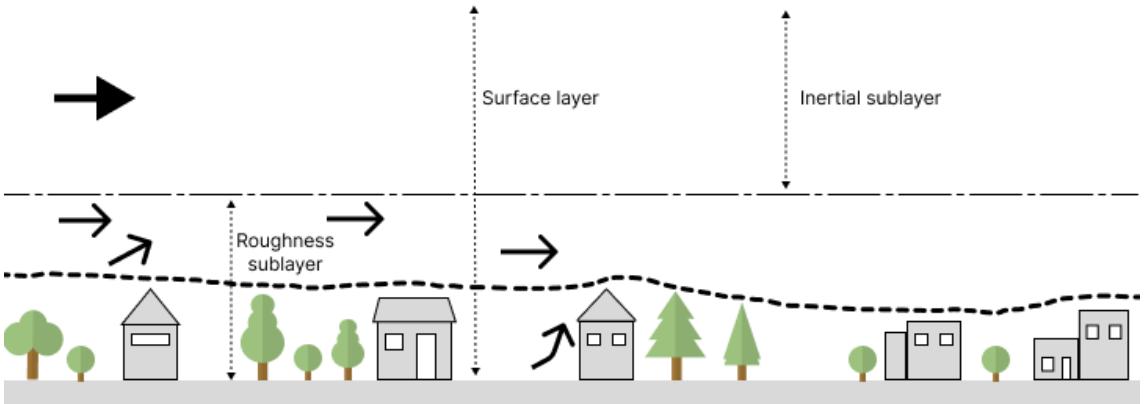


Figure 2.2: Local scale view of the urban climate, adopted from [Oke06]

The local scale is situated closer to the surface and contains landscape features such as topography but does not yet include microscale effects. At this layer, the underlying microclimatic effects in form of fluxes mix to form a more average and representative view of the source area, typically at the scale of one to several kilometres. This layer is monitored by weather stations that are located at/or slightly above the canopy height.

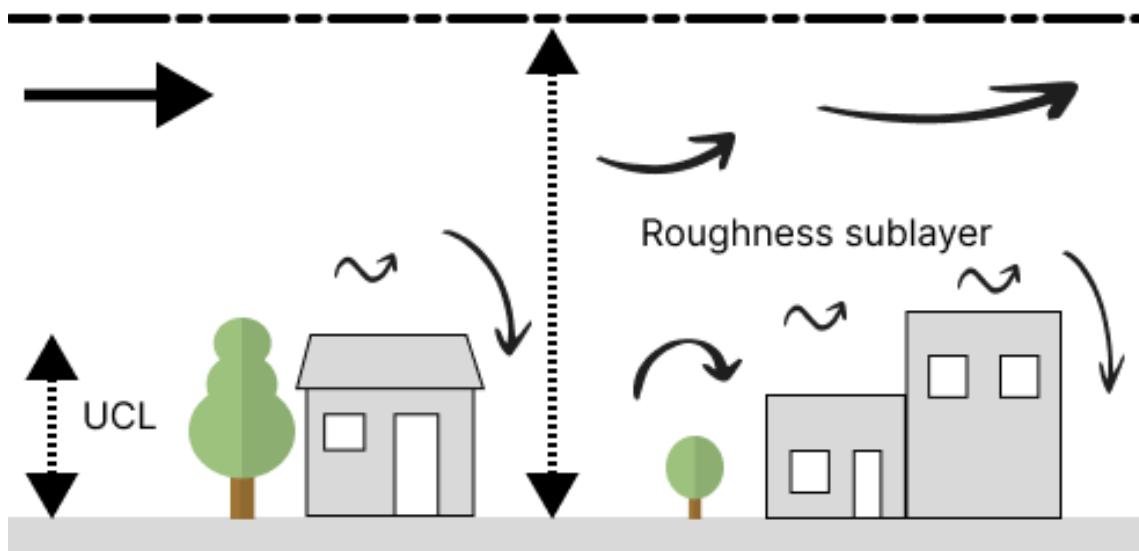


Figure 2.3: Microscale view of the urban climate, adopted from [Oke06]

The microscale usually ranges from a neighbourhood scale to individual street canyons or even microclimates created by individual buildings. It is mainly influenced by the overall energy balance, which is influenced by cloud coverage, solar radiation and more. Single weather stations are not enough to capture the complex microscale [Oke04], therefore a dense network of sensors closer to the ground is needed to capture the microscale. Additionally, the microclimate is influenced by LST; however, the correlation between air and LST varies greatly based on other surrounding influences such as wind velocity or humidity [SB92], especially in cases of extreme temperatures [Goo16]. The closer the  $T_{air}$  is measured to the surface, the more local the measurement, therefore  $T_{air}$  of the canopy is usually measured at 2m height, so the different energy fluxes have enough time to mix and form a more representative average for a bigger area.

Vertically, UHIs can be divided based on these boundary layers into three major types [Oke76, OMCV17], namely Boundary Layer Urban Heat Island (BUHI), Canopy Urban Heat Island (CUHI), and Surface Urban Heat Islands (SUHI), corresponding to the boundary layer they can be measured in. Another differentiating factor is the time of day at which UHIs get detected. For example, Steward [Ste11] in his review focussed on night-time UHIs, whereas Peng et al. [PPC<sup>+</sup>12] focussed on both day and night-time UHIs.

## 2.1.2 Surface Urban Heat Island

The LST is measured directly at the surface of an object and is the main indicator of the SUHI, which can be found in many cities around the globe [PPC<sup>+</sup>12]. LST, in contrast to  $T_{air}$ , is measured via remote sensing technologies via satellites. Well-known satellites include MODIS [Did21] (NASA), Sentinel 3 (ESA), Meteosat (EUMETSAT), Landsat, and more. The different satellite types carry different types of instruments and sensors, that

can take various measurements. The used sensor has a major influence on the quality of data, e.g., resolution via pixel size, robustness against atmospheric influences, ability to handle clouds, and more. In Section 4.5 we discuss other features next to LST that are available via satellite data.

Through the use of satellites, the spatial coverage is great, but raster sizes for older satellites such as MODIS usually range from one to several kilometres, therefore the spatial resolution is not as high. For newer satellites such as Sentinel 2 and Landsat, pixel sizes are improved significantly from 10m to 50m; however, complementary data such as derived vegetation indexes are usually not available as precomputed values, adding additional work to the data retrieval process, as later discussed in Section 4.3. Additionally, weather satellites usually orbit earth to cover wide areas and are not geostationary, therefore only taking measurements a maximum of 1 to 2 times a day, up to every 16 days or even only once a month, depending on the orbit. As a result, the temporal resolution is quite low and especially in the case of UHI detection, this could mean that the satellite misses the peak of the UHI for a given day or even misses the UHI completely. Another downside is the general inability to take LST measurements through clouds, therefore even if the satellite passes over during a UHI, if there are clouds, the UHI cannot be measured. To alleviate the problem there are new methods such as estimating LST based on emitted radiation from clouds [ZPL15]; however, they also have their limitations. Lastly, LST and  $T_{air}$  are not the same and can vary greatly, especially in extreme heat events [Goo16].

To conclude, SUHI analysis can be a good indicator that there is a UHI phenomenon present in a city and can generally direct further research; however, it lacks the temporal resolution to be used for real-time UHI detection and is not able to capture microscale effects of the UHI [VO03, VS17].

### 2.1.3 Canopy Urban Heat Island

The CUHI is measured in the canopy boundary layer several meters above ground slightly below or on the average roof layer of the surrounding buildings, as seen in fig. 2.3. The primary measurement in the canopy is  $T_{air}$ , which is used to measure the urban heat island intensity (UHII) [Oke73], the most used way of describing the heat island magnitude [KB21].

Since the beginning of modern climatology, major progress has been made in this research field, but methodologies and scientific rigor in CUHI research still seems to be lacking, as discussed by Stewart in 2011 [Ste11]. Stewart found, that over 54% of CUHI research was lacking proper methodologies or had other shortcomings such as a lack of site descriptions, where sensors were placed, or the disregard of non-urban factors such as local weather phenomena. In response, progress has been made in recent years by improving methodologies and ensuring correct measurements of climate-related data and study design and execution through various guidelines [Oke06], especially in urban set-

tings, that require special care due to the huge number of possible influences on local recording sites.

Additionally, the UHII is highly related to other climatic factors such as wind, cloud cover, and precipitation and is tightly linked to the selection of the recording site [FHM<sup>+</sup>19], therefore such factors need to be considered when measuring the UHII.

Compared to LST,  $T_{air}$  is measured *in situ* via weather station networks or other types of sensor networks. Provider for PWS and temperature sensor networks are further discussed in Section 4.1.

### Shared UHI Challenges

Some shared challenges for all types of UHI include:

1. Define what *urban* means in the context of UHIs [SO09]. The term urban is widely used to identify areas that are more densely populated than the surrounding rural areas. Having this distinction between urban and rural [Low77] helped researchers to better define the UHI magnitude, but this simple distinction also leads to problems [Ste11]. The problem lies in the fact that there is no clear border between urban and rural areas, but a fluent transition. Especially for larger metropolitan areas, like Tokyo, the urban area could span 10s to 100s of kilometres, making the collection of reference rural temperatures hard. The reference rural temperature has a direct influence on the UHI magnitude, which is ‘the most widely recognized indicator of city climate modification in the environmental sciences’ [SO09]. As a solution, different classification into local climate zone (LCZ)s were proposed [SO12, SO09], that classify areas based on surface roughness, building densities, building heights etc.
2. Measuring the influence of other local urban or meteorological phenomena on the temperatures collected. The urban climate is extremely complex, due to many different influences, such as anthropological energy, heat dissipated from ACs, vehicles, and more. Additionally, the urban climate is also influenced by surrounding regional/meso-scale climate phenomena such as storms, valleys, mountains, large waterbodies, coastlines, and more. In Section 4.5, we talk about potential features to capture the local urban climate.

## 2.2 Smart Cities

Smart Cities offer many new possibilities, enabling new ways of communication and sensing applications. To find out how Smart Cities are structured and how applications could take advantage of data provided by a Smart City, we look at its general architecture. The most generalised architecture of a smart city consists of four layers, the sensing layer, transmission layer, data management, and application layer [SKH18]. In this work, we focus on the sensing layer, dealing with topics such as correct sensor placement and underlying sensor footprints, and the application layer, which accesses available data via data management services to provide additional services to the city and its citizens.

For the data transmission and data management layers, there already exist different technologies and service offerings, that aim at solving the underlying problems, e.g., network bandwidth, network availability, sensor discoverability, handling the massive amounts of data that is already or will be collected in the future, and many more. For the communication and discovery of sensor nodes, one solution could be SkipNet [HDJ<sup>+</sup>02], an overlay network focused on discoverability while also protecting privacy, with which the data transportation layer could be designed as a peer-to-peer (P2P) network. Other research focuses on the data accessibility and discoverability, by making data accessible for everyone, not only for economic partners in a closed-off system. Examples would be the Smart Networks for Urban Citizen Participation (SANE) initiative [BJK<sup>+</sup>19], which could provide crowd-sourced distributed  $T_{air}$  sensing with a framework to make sensors searchable and subscribable, allowing real-time applications by consuming sensor data streams. Figure 2.4 shows how an architecture with SANE could look like.

In connection to this work, ML-based interpolation, for example for  $T_{air}$ , could be used to augment Smart City applications by supplying data for sensor locations while a sensor is offline, or by turning discrete sensor readings into a temperature map that could be used by researchers for further heat related analysis or by decision makers to inform and visualise heat stress in a city.

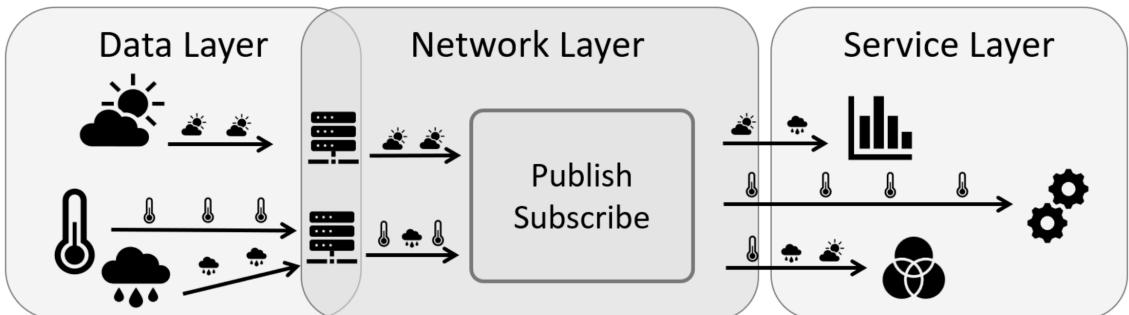


Figure 2.4: In the data layer (left), a wide variety of environmental data is collected with the help of multiple sensors. These are connected to their citizen-owned local base stations, which manage access rights and forward collected data to subscribed services (right) via the decentralized publish-subscribe in the network layer (centre).

### 2.2.1 Sensing Layer

The goal for the sensing layer is to monitor the surrounding environment and capture key data for further analysis and decision making. It consists of many different types of physical and virtual sensors. The first group of sensors are the physical sensors, which are placed directly inside the environment. Wireless sensor networks (WSN) [DP10] have seen a lot of attention for many different applications such as ‘military sensing, physical

security, air traffic control, traffic surveillance, video surveillance, industrial and manufacturing automation, distributed robotics, environment monitoring, and building and structures monitoring' [CK03]. The challenges for WSNs primarily depend among other things on the deployment. An ad-hoc WSN has energy and bandwidth constraints due to the usage of batteries as power sources. In contrast, sensors that are permanently installed, either stationary or on a moving target, and connected to a constant power source don't have these constraints. This approach could be used for Smart Cities to reduce waste and guarantee representative measurements via correct sensor placement. In the case of stationary sensor networks though, the initial deployment and following maintenance cost can be substantial, as seen by the Birmingham Testbed [CMY<sup>+</sup>15]. If the cost is however distributed across many by following a crowd-sourcing approach, e.g., crowd-sensing via PWS, larger networks could be maintained, but there are also new challenges introduced by running sensor networks by non-professionals [MFG<sup>+</sup>17].

In recent years, low-cost sensors (LCS) in combination with sensor networks have enabled fine-granular real-time monitoring of urban environments [Gri06, RGA<sup>+</sup>09], although the quality of individual LCSs can be questionable [CDS<sup>+</sup>17]. Especially PWS data has been used to augment weather data from traditional weather monitoring networks from official weather services [HGMS<sup>+</sup>22]. In general, LCSs can improve data availability and support analysis, but do not substitute well-calibrated reference instrumentation [LPS18].

### Stationary Sensors

There are many different types of environmental features that can be measured directly inside an urban area. The types of measurements that can be observed are among others:  $T_{air}$ , humidity, atmospheric pressure, reactive gaseous air pollutant (CO, NO<sub>x</sub>, O<sub>3</sub>, SO<sub>2</sub>), particulate matter (PM), greenhouse gases (CO<sub>2</sub>, CH<sub>4</sub>), precipitation, solar radiation, wind speed and direction, anthropogenic heat, noise, sky-view factor (SVF), heat fluxes, and many more. Correlations between these features can vary greatly based on surrounding factors. To better understand these correlations, many empirical studies have studied the influence of meteorological factors on features such as PM [TMJ10]. Additionally, many fields of statistics have specialised on topics such as statistics in climatology [VSZ02], geostatistics [TYU86] and more.

All sensor readings that are taken by physical sensors are singular data points. Additionally, to the type of measurement taken and the actual value observed, physical sensor readings include the physical location of the sensor, e.g., latitude, longitude and sometimes altitude, the type of sensor used to take the measurement, and sampling rates.

For  $T_{air}$ , the sampling rate could be an average temperature measured over five minutes, whereas precipitation might be measured by collecting rain for certain periods of time and then measuring the amount of rain collected. The sensor type is important, as different types of sensors can produce different qualities of measurements, e.g., LCS

have lower accuracy compared to calibrated reference-grade high-cost sensors and might have lower response rates, and perform better or worse based on the meteorological conditions, e.g., worse performance at low temperatures, high humidity etc. Due to the placement directly inside the environment, (near) real-time observation and high temporal resolution are generally possible but might be influenced by factors such as network availability. The spatial resolution highly depends on the number of sensors deployed and the correct placement of the sensors. The correct placement has a direct influence on the footprint of the sensor [LF14] and the representativeness of the measurement taken for the underlying and surrounding area [Oke06].

One downside of the placement directly in the environment the sensors are observing, is the exposure to environmental influences such as heat, humidity, or pollution, that can decrease the lifetime of a sensor and may require more frequent maintenance or replacement. An example could be that due to pollution in the air, a rain gauge might be cleaned/replaced more often as over time dirt builds up very quickly.

### Mobile Sensors

Next to stationary sensors such as a PWS, mobile sensors are not bound to one place, but have the ability to move through the environment they are placed in. This increases the spatial coverage at the cost of temporal resolution, as a sensor is not always in the same place. In the context of urban  $T_{air}$  sensing, LCS could be mounted to moving vehicles such as busses, cars, e-bikes, scooters, and more. This could help improve the spatial coverage of a sensor network. In combination with ML-based interpolation, such moving sensors could be used to create virtual sensors for specific locations, where a ML model learns the relationship between surrounding sensors so that  $T_{air}$  could be interpolated for times when a moving sensor is currently not at the specific location. In their study, Yang et al. [YBZ19] found that randomly mobile sensor networks outperformed completely stationary sensor networks for measuring monthly mean temperatures; however, they had higher errors of up to 5°C when measuring daily maximum temperatures. They suggest that hybrid systems with both stationary and moving sensors are more robust in measuring short extreme events such as heat waves.

Next to mobile sensor networks, official weather services such as the DWD also do temperature profile measurements<sup>1</sup>, where sensors are mounted to a car to capture  $T_{air}$ , humidity, wind speed and direction, as well as possibly atmospheric pressure. This data can be used to support research into local climates. Especially interesting for these profile drives are ‘summer cloud-free and light-windy high-pressure weather conditions, as then temperature differences between the city and the surrounding area become particularly pronounced and temperature-equalising cold air flows reach their greatest intensity.’<sup>2</sup>.

<sup>1</sup><https://www.dwd.de/DE/service/lexikon/Functions/glossar.html?lv2=101996&lv3=102106>, last accessed: 23.08.2023

<sup>2</sup>[https://www.dwd.de/DE/forschung/klima\\_umwelt/klimawirk/stadtpl/projekt\\_warmeinseln/projekt\\_waermeinseln\\_node.html](https://www.dwd.de/DE/forschung/klima_umwelt/klimawirk/stadtpl/projekt_warmeinseln/projekt_waermeinseln_node.html), last accessed: 23.08.2023

### 2.2.2 Application Layer

The application layer contains services which utilize data provided by the data management layer to provide services for the city and its citizens. As part of this layer, services could be built that aggregate data streams coming from the data management layer and use ML to improve the data quality by detecting outliers, reducing bias, interpolating missing data etc. The improved data could then be published and other services that would otherwise rely on the raw data streams and potentially need to implement their own outlier detection or interpolation of missing data techniques, instead simply subscribe to the externally managed service. This could lower the barrier to entry for developers with less available resources, financially or domain-knowledge wise, and generally allow developers/service providers to allocate their resources to other areas like user experience (UX) and usability compared to the maintenance of complex ML-based services. In the context of this work, we focus on  $T_{air}$  interpolation and have the motivating factor of UHI detection. In this context, there could be a UHI detection service that ingests real-time data streams from the data management layer and notify citizens if an UHI is detected in or predicted for a particular urban area. As the main challenge for UHI detection lies, next to the definition of urban and rural reference areas, on the gathering of a comprehensive temperature map that allows for UHI detection algorithms to work, in this work we also evaluate areal interpolation of  $T_{air}$ . Such areal interpolation techniques could then be used to create a temperature map service that enables the detection of CUHIs and could also be used as a foundation for other services in a smart city, like plant watering systems, smart healthcare and more. Examples for (crowd-sourced) temperature maps are later shown in Section 4.1, as many PWS providers also operate a temperature map as well. The problem here is that every provider has its own temperature map with custom ways of storing and accessing the data, making it difficult and time consuming to work with different providers.

## 2.3 Interpolation

Interpolation is in essence to determine unknown data points based on a set of given data points [Ste27]. In this work, we focus on spatial interpolation which is the interpolation problem applied to spatial data given either as discrete data points or subareas, to determine a complete area. First, we categorize and introduce different spatial interpolation methods to get an understanding about what methods exist, which method is preferred in which application area based on the literature review by *Lam* from 1983 [Lam83], and augment certain areas with the current state of research. In recent years, many more specialised interpolation methods have been developed for specific use cases, as interpolation is not only about randomly selecting values, but estimating data points based on assumptions about the relationship with the existing data points and the area to interpolate. Depending on the exact use-case, these assumptions could be about the distribution

---

of the data, or as a specific example in the case of interpolating liquids, having constraints such as that method is volume-preserving.

Generally, spatial interpolation methods can be categorised by many different factors. *Lam* differentiated between point interpolation, which is either exact or approximate, and areal interpolation, which is either non-volume-preserving and the same as point-based interpolation, or volume-preserving. As *in situ* sensor readings are singular data points at discrete locations, we focus on point interpolation, but note here that such data points could also be mapped into a partial grid first to then use areal interpolation methods. Point interpolation methods can either be exact, meaning that the original data points are preserved 1 to 1 in the interpolated area, or approximated, e.g., they are fitted to a function that does not necessarily pass through all original data points. The important methods are as follows:

- Exact
  - Weighting
  - Kriging
  - Splines
  - Interpolating Polynomials
  - Finite Difference
- Approximate
  - Power Series Trend
  - Fourier Series
  - Least-squares Fitting with Splines
  - Distance-weighted Least-squares

### 2.3.1 Exact Point Interpolation

Exact point interpolation methods have the benefit that the original data points are preserved. Fitting the original data points to a polynomial is the simplest form of interpolation; however, it has the major drawback that there are no additional constraints for points that are not part of the original data set, potentially resulting in highly unreasonable estimations.

The main idea of weighting methods is the idea to assign more weight to data points that are closer than points that are further away. Due to its simplicity and fast computation, inverse-distance weighting (IDW) [WRP85] is a popular and commonly used interpolation method. The main downside with IDW is that it is a smoothing technique, therefore it is not able to capture local maxima and minima, which could be critical for UHI detection. Splines [MM88], another exact smoothing technique, is a mathematical method that

fits either a smooth curve or surface to a set of given points. It offers several advantages such as smoothness and retention of small-scale features; however, this method can be computationally expensive and might not be best suited for highly irregular data points, e.g., big differences between data points located very close to each other. Finite Difference is a method to calculate a surface based on a set of differential equations, which can calculate a smooth surface from the given points; however, at the cost of higher computational cost to solve the differential equations iteratively.

Kriging, a geostatistical interpolation method, originally developed by Krige [Kri76] as a moving averaging technique to reduce global biases, has developed into one of the most prominent spatial interpolation methods and has seen many contributions and improvements since the introduction of Ordinary Kriging (OK) and Universal Kriging [LH14].

## Kriging

Due to its popularity, we discuss Kriging methods in more detail. Kriging methods use a covariance function to model the spatial correlation between data points [Wac03]. The covariance function is a measure of the similarity between two data points, which is used to calculate the weight of the data point in the interpolation process. There are different types of Kriging methods available, each suited for different use cases, as offered by ArcGIS<sup>3</sup> including:

1. Simple Kriging: the simplest form of Kriging, that assumes that the mean of the measured values is known and constant.
2. Ordinary Kriging: same as Simple Kriging but the mean is an unknown constant.
3. Universal Kriging: instead of assuming a constant mean, the mean is modelled as a deterministic function.
4. Indicator Kriging: same as Ordinary Kriging but for categorical data.
5. Probability Kriging: same as Indicator Kriging but assumes two types of random errors that can be each auto-correlated and cross-correlated to each other.
6. Disjunctive Kriging: same as Ordinary Kriging but tries to improve the prediction quality by using an unknown constant and approximating an arbitrary function. It requires the bivariate normality assumption and is difficult to verify and solutions might be mathematically and computationally complicated.
7. Cokriging: offers methods for the previous Kriging methods but uses information on several variable types. This could improve the prediction quality, but might increase the variance of the prediction, as more much more estimation is required.

<sup>3</sup><https://desktop.arcgis.com/en/arcmap/latest/extensions/geostatistical-analyst/what-are-the-different-kriging-models-.htm>, last accessed: 24.08.2023

In the context of geostatistical analysis, there are different types of Kriging methods available that combine the aforementioned methods with other techniques, such as regression analysis. The following list is the geostatistical methods offered by ArcGIS Pro as part of the Geostatistical Analyst toolbox<sup>4</sup>:

1. Empirical Bayesian Kriging (EBK)
2. Empirical Bayesian Kriging 3D (EBK3D)
3. EBK Regression Prediction (EBKRP): Empirical Bayesian Kriging with regression prediction
4. Global Polynomial Interpolation
5. Kernel Interpolation with Barriers
6. Moving Window Kriging
7. Radial Basis Function

ArcGIS Pro is a paid service therefore we only take a look at openly available implementation, more precisely the Kriging implementation from the Python library *PyKrig* [MYM22], which are the following:

- Ordinary Kriging
- Universal Kriging
- Regression Kriging

---

<sup>4</sup><https://pro.arcgis.com/en/pro-app/latest/tool-reference/geostatistical-analyst/an-overview-of-the-geostatistical-analyst-toolbox.htm>, last accessed: 24.08.2023

### 3 Machine Learning-based Interpolation

In recent times, the area of machine learning has seen big advancements in terms of model size and complexity. Especially in the area of generative artificial intelligence (AI), transformer-based neural networks have revolutionised text and image generation. Models such as OpenAI's *ChatGPT* [Ope23] or Google's *LaMDA* [TDFH<sup>+</sup>22] have generated significant hype for the possibility of use of AI. Additionally, statements like the universal approximation theorem, which states that a feed-forward network with a single hidden layer containing a finite number of neurons can approximate any continuous function [HSW89], emphasize the potential power of ML models. As a result, the question arises what benefits AI can bring to other areas of application, such as interpolation. In this chapter, we will discuss the usage of ML in the context of data enrichment via interpolation, more precisely in the context of Smart Cities and urban  $T_{air}$  interpolation. The ML models will be compared in Section 5 to traditional proven geostatistical model, e.g., Kriging, to outline and discover possible advantages and disadvantages. In general, the idea is to trade the explain- and interpretability of purely statistical-based approaches for model capabilities and accuracy, and the ability to capture more complex (non-linear) dependencies. Due to the great flexibility of ML models, each model can be fit to completely different use cases, such as interpolation vs. extrapolation or areal interpolation vs. interpolation of a single location. The following sections introduce different ML models and discussed their applicability to the use-case of urban  $T_{air}$  interpolation.

#### AI vs. Machine Learning vs. Deep Learning

Before diving deeper into the applications of ML, we need to clarify what is meant by AI, (ML) and deep learning (DL). AI is a broad term that is used to describe the ability to perform tasks, that are usually associated with human intelligence. ML is a subfield of AI, that focuses on the ability of a system to learn from data without being explicitly programmed to do so. Finally, DL is a subfield of ML, that uses Artificial Neural Network (ANN)s, or also called Simulated Neural Network (SNN), which imitate the structure of the human brain, to learn from data and perform various tasks.

---

### 3.1 Machine-Learning Application Areas in Air Temperature Interpolation

As meteorological research and analysis activities are usually in need of gridded or continuous data [SKP<sup>+</sup>20], interpolation is a really important tool to convert single data points at distinct locations from ground-based weather stations into a continuous layer. Interpolation can also be applied to individual sensors to fill in missing data that are caused by network outages or to increase the temporal resolution to turn hourly into sub-hourly readings. Especially with the capability to increase the temporal resolution, the question arises how this capability could be combined with moving sensors to increase the spatial coverage of a sensor network. In this work, we will discuss two approaches for the use of interpolation in urban  $T_{air}$  sensing:

- ML-based interpolation for areas as a substitution for geostatistical methods, e.g., Kriging, to turn individual data points into a continuous grid, possibly with the ability to handle stationary and moving sensor data at the same time
- ML-based interpolation for a single location to interpolate timeframes with missing data or to simulate a higher temporal resolution that does not only interpolate between individual data points of the same sensor, but also takes into consideration surrounding sensors

Research suggests that for fine-granular spatio-temporal urban  $T_{air}$  maps, a sensor density of at least 1 sensor per  $\text{km}^2$  is needed [VBEM20]; however, the denser the sensor network the better the prediction quality, as even inside a single street canyon  $T_{air}$  can easily vary by 2 to 3°C [SHN<sup>+</sup>08]. To achieve this sensor density as well as to gain insights into previously unobserved areas and to minimise prediction uncertainty for those areas, hybrid approaches combining stationary and moving sensors have shown to work better than purely stationary networks by covering more ground as well as reducing variability of purely mobile network setups in the context of urban temperature sensing [YBZ19]. The combination of reference grade stationary sensors and moving sensors also shows promise in other related application, e.g., in the context of pollution island detection [IBA<sup>+</sup>22].

In the context of this work, we discuss both ML applications and try to show the feasibility and potentials of ML-based interpolation in the context of urban  $T_{air}$  sensing. In the following, we introduce several ML algorithms that can be used for regression tasks and discuss how they need to be adapted to solve interpolation tasks as well as the advantages and disadvantages of each model. The models will be implemented as prototypes and evaluated in Chapter 5.

## 3.2 Model Selection Criteria

Before using the ML regression algorithms introduced in this section to solve the interpolation problem, the models need to be adapted to this specific use-case. This can happen either by adapting the input data and the types of features used or by adapting the model configuration. The following questions need to be answered:

- **How to model sensors:** Are sensor locations modelled individually, as a network, or as a grid?
- **How to model the temporal correlation:** Does the model allow to model temporal correlation between sensor readings and is it only short-term or also long-term correlation?
- **How to model the spatial correlation:** Is spatial correlation directly incorporated in the model architecture or does it need to be modelled via features?

Next to the adaptations that need to be made to fit regression algorithms to the interpolation problem, there are also non-functional requirements that need to be considered when selecting a model. The most important requirements are:

- **Model Assumptions:** The model assumptions need to be met by the features used in the input data. For example, linear regression assumes that the input features are independent from each other, as linear regression measures the amount the target variable is influenced by one feature changing while all other features stay the same. In case of correlation, this assumption is violated, as for example the amount of precipitation influences the humidity. Other assumptions could be made about the distribution of data, the mean of values, etc.
- **Accuracy and Reliability:** Creating an accurate and reliable model is important to increase the trust for predicted values; however, there are certain trade-offs to be made, as model performance or generalisation ability are also important factors to consider. The accuracy of the model is mainly determined how well the chosen model can fit the underlying data, e.g., a linear model cannot fit a non-linear function, and is measured by the evaluation metrics described in Section 5. The reliability of the model is determined by the training data as well as the model architecture, as for example training data that is not representative of the underlying function can introduce bias into the model or can prevent the model from learning the correct function. Another important factor is the data quality, as more noise can result in worse model performance. Lastly, reliability of the model is also determined by the ability of the model to handle missing or sparse data as well as outliers. This is especially important in our context as we try to integrate moving sensors into the interpolation process, which sense data at different times and locations.

- **Extrapolation Capabilities:** One important factor to consider, especially when using a model with previously unseen data, is the ability to extrapolate data. Linear regression models as an example try to fit a linear function that can be easily used with data that is bigger/smaller than previously seen training data as the function is continuous. In comparison, regression trees by default do not allow for extrapolation. Also, ANNs typically perform unpredictably on unseen data.
- **Amount of Training Data:** The amount of training data required to train the model is another important factor to consider, as some models require more training data than others. Especially neural networks tend to need more training data than other models, as they have lots of parameters that need to be tuned. In our context, the amount of data available is quite limited, therefore models that require less training data are preferred.
- **Handling of Missing Data:** If the model cannot handle missing data well, there might be additional data pre-processing steps that need to be done. One example for this would be how the model reacts to not a number (NaN) values which is a float number defined in the IEEE 754 floating-point standard [iee19]. Each multiplication with NaN results in NaN, which can therefore lead to a model where all weights turn into NaN when there is a single NaN value in the input data. This is especially important in the context of distributed sensors, as they might not sense every feature at all times. Common strategies to handle missing values involve dropping the complete feature if it has any NaN, drop any rows in the data that has NaN values or imputation, e.g., replace the missing value with a value such as the mean or median of the feature.
- **Handling of Sparse Data:** Like missing data, handling of sparse data is also important to prevent problems such as the NaN problem mentioned previously. The main difference to missing data is that sparse data is not missing, but rather not always available. In our context, moving sensors would be an example for sparse data, as they only sense data at certain times and locations. The strategies to handle missing data are also similar to those of missing data, but imputation and interpolation are more common strategies to handle sparse data.
- **Model Performance:** The more complex a model is, the more training data is needed to fine-tune all its weights and the longer it takes to train, either due to the amount of data or the number of steps that need to be taken when updating weights in the training process. In the context of open-source and citizen participation less complex models are preferred, as they can be trained and deployed with less resources. However, a less complex model could have the downside of not being able to fit the underlying data as well as a more complex model, therefore there is a trade-off between model complexity and model performance. Another factor is

the capability of handling many locations and data points, as some models' performance degrades significantly when the amount of data points increases, as well as the ability to handle high-dimensional data.

- **Model Capabilities vs. Interpretability:** A common trade-off in ML models is between model capabilities and complexity and interpretability of the model as done by [ZKBK21], as 'Neural network and deep learning approaches allow for large flexibility and predictive power but are harder to interpret than ensemble-based approaches which allow for the required flexibility, while still providing insights into the algorithms' inner workings, e.g., via variable importance and prediction uncertainty estimation'.
- **Other:** Next to these main requirements, there are also other requirements, such as the ability to handle massive amounts of data, live retraining, or sophisticated support via ML libraries such as *Tensor Flow*<sup>1</sup> for commercial use cases. Due to the limited scope of this work, these requirements will be considered in less detail.

After introducing the requirements for model adaptions and selection, the next step is to introduce and compare the different ML regression algorithms. Generally speaking, ML algorithms can be categorised based on many different properties [Sar21], such as the type of learning, e.g., supervised, unsupervised, semi-supervised or reinforcement, or the type of problem they try to solve, e.g., classification, regression or clustering. The most important differentiation for this work is to distinguish algorithms based on the type of problem they try to solve. Because we focus on solving interpolation problems, in this work we will only consider algorithms that can be used to solve regression problems, as interpolation is a form of regression.

In regression analysis, the goal is to predict a (continuous) target variable  $y$  based on a set of input variables  $X$ , like it is the case for temperature interpolation. This problem can be classified as a supervised learning problem, therefore possible algorithm candidates contain the following models:

- (Multi) Linear Regression
- K-Nearest Neighbours Regression
- Regression Trees and Forests
- Histogram-Based Gradient Boosting
- Support Vector Regression
- (Deep) Neural Networks

---

<sup>1</sup><https://www.tensorflow.org/>, last accessed: 01.09.2023

Next to these algorithms, there also exist less popular interpolation methods, such as outlined in [LH14]; however, due to the limited scope of this work, we only take a look at ML models listed above, who are implemented in the popular *sklearn* ML library [PVG<sup>+</sup>11]. Each model has certain benefits but also comes with drawbacks or special assumptions for the input data to the model. First, we will discuss each model and then compare these assumptions with the data coming from the data-layer, to identify suitable models for the task of  $T_{air}$  interpolation. Based on the domain, there already exist proposed best practices for which algorithms to use for what applications. In the context of Smart Cities such recommendations exist for topics such as intelligent transportation systems, smart grids, smart city health care and more can be found in [UATMG20], which does not cover interpolation. In the context of  $T_{air}$  interpolation, regression forests and HistGB seem to be popular choices and perform better compared to other methods [AYDK22, HKS<sup>+</sup>14].

### 3.3 Comparison of Machine Learning Algorithms

#### 3.3.1 (Multi) Linear Regression

Linear regression [MPV21] is a comparatively simple, yet very powerful and widely used model for regression problems. The goal of this model is to predict a continuous dependent variable based on several independent variables. These independent variables can be either continuous or discrete. The model can be expressed as follows:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \epsilon \quad (3.1)$$

where  $y$  is the dependent variable,  $x_1$  to  $x_n$  are the independent variables,  $\beta_0$  to  $\beta_n$  are the parameters of the model and  $\epsilon$  is the error term. The relationship between the parameters is assumed to be linear, while each variable must be independent of each other. Due to this independent assumption, there are special steps needed to make linear regression work for (geo-) spatial data, as these types of variables are usually correlated with each other. This is further discussed in 4.6.

Linear regression has the advantage that it is a very simple model, therefore most of the work needs to be done in the feature engineering process. The downside is that there is no inherent support for spatial or temporal correlation and the model cannot be used to fit non-linear functions. To fit non-linear functions, polynomial regression can be used, which can however lead to overfitting especially when the degree of the polynomial is high.

Linear models seem to perform worse in urban temperature related settings, as they are unable to capture non-linear effects [VS17], therefore this model will only be evaluated briefly.

### 3.3.2 KNN Regression

K-Nearest Neighbours (KNN) is a simple algorithm that can be used for both classification [CH67] and regression problems [Alt92]. The main idea behind KNN is the assumption, that data points near each other are more similar than data points that are further away. As a result, the  $k$  nearest neighbours, either by number or by radius, of a data point are used to predict the target variable. The number of nearest neighbours is a hyperparameter that needs to be tuned to find the best trade off between bias and variance. The model with equal weights per neighbour can be expressed as follows:

$$\hat{y} = \frac{1}{k} \sum_{i=1}^k y_i \quad (3.2)$$

where  $\hat{y}$  is the predicted target variable,  $k$  is the number of nearest neighbours and  $y_i$  is the target variable of the  $i$ -th nearest neighbour. Additionally, weight could be assigned to individual neighbours to for example weight closer neighbours higher.

KNN is part of the family of non-parametric models, meaning they do not make any strong assumptions about the underlying regression curve. KNN is a simple yet powerful model and based on the weight function, all predictions can either be weighted equally, by distance, or by a custom function, that allows potentially more complex weight calculations.

### 3.3.3 Regression Trees and Random Forests

Tree predictors are used for a wide variety of classification and regression problems. Since tree predictors are unstable, e.g., vary significantly given similar inputs, and tend to overfit, random forests were introduced as a counter measure. Random Forests (RF)s combine multiple tree predictors and train them on different features and subsets of data and either average their predictions to reduce the variance or use boosting methods to reduce the bias of a combined estimator [Bre01].

In the case of predicting a continuous target variable, regression trees can be used. The principle behind regression trees is to split the data into continuously smaller sub-sets and organise the splitting points in a way that minimises the error. Compared to decision trees which try to minimise the entropy, regression trees try to minimise an error that is compatible with a continuous target variable such as mean squared error (MSE). Regression trees are comparatively easy to understand and interpret and offer certain benefits such as feature insensitivity, meaning that features do not need to be scaled before usage and can be used as is. In [ZKBK21] Zumwald et al. choose to use Quantile Regression Forests (QRF) [MR06] for mapping hyperlocal  $T_{air}$  in Zurich, Switzerland, due to the flexibility and predictive power of ensemble-based approaches and the ability to still gain additional insights into the algorithms' inner workings via variable importance and prediction uncertainty estimation. They note however that ANN and DL approaches allow for even larger flexibility and predictive power but lack behind in other areas such

as the aforementioned interpretability. In the following we get an overview of RMSE and  $R^2$  values achieved by related studies.

Ho et al. [HKS<sup>+</sup>14] mapped maximum daily  $T_{air}$  for Vancouver, Canada on hot summer days and combined remote sensing TM/ETM data from Landsat with field observations from Environment Canada and WOW. They compared Ordinary least squares regression, Support Vector Machine (SVM) regression, and RF Regression and achieved the following RMSE:

- RF: RMSE 2.31°C
- SVM: RMSE 2.46°C
- Ordinary least squares regression: RMSE 2.46°C

They also added that stations closer to the ocean had higher estimation errors, possibly due to more variable wind patterns [RO00].

Hengl et al. [HNW<sup>+</sup>18] adapted the RF algorithm to take advantage of spatial and temporal variables, called Random Forest for Spatial Predictions Framework (RFsp). They compared the following ways of modelling spatial relations:

- Geographical coordinates (north, east)
- Euclidean distances to reference points (centre, edges) of the study area
- Euclidean distances to sampling locations, e.g., distance matrices also used in geostatistics
- Downslope distances for hydrological analysis
- Resistance distances or weighted buffer distances

They found that only using coordinates resulted in lower  $R^2$  and higher MSE errors compared to distances between sampling locations, and concluded that RFsp performed similar to OK; however, without the downsides of needing to create and fit a variogram, no manual trend building, no need to define a search radius for Kriging, no transformation of the target variable, spatial autocorrelation and correlation with spatial environmental factors are modelled together, and variable importance shows which observations and covariants have the most influence on the predictions. They described the advantages of RF as follows:

- Information overlap (multicollinearity) and over-parametrization are not a problem. RF can even be trained with more covariants than observations.
  - RF has built-in bagging methods to select subsets of features and training samples to reduce bias. In geostatistics over-parametrization and overlap of covariants are an issue and can lead to bias predictions.
-

- noise-resistance [SBZH07]
- Distance measures can be extended to more complex calculations

The downsides of RF according to them include:

- Extrapolation can lead to even poorer results compared to linear models.
- If the training data is biased, the RF will have biased predictions. Therefore, stricter cross-validation techniques might be necessary.
- Model size is much larger compared to linear models.
- Model is optimized for representing the training data, not the spatial and spatiotemporal dependence structure.
- Computationally expensive to train predictor and make predictions.
- Distance matrices scale quadratically, therefore higher number of points and greater radiuses/number of neighbours result in way higher computational cost.

Lastly, they also concluded, that the reduction of trees in the forest from 500 to 100 often didn't result in a big loss in accuracy, suggesting that simpler RF models can retain high levels of accuracy.

*Sekulić et al.* further improved RFsp and introduced Random Forest for Spatial Interpolation (RFSI) [SKH<sup>+</sup>20] by including nearest observations and distances as covariants. They compared their model performance against OK, RFsp, IDW, NN, and trend surface mapping [CH65] to compare the performance across one synthetic dataset, one precipitation dataset, and one mean daily temperature dataset for Croatia 2008 with a resolution of 1km<sup>2</sup>. In the real case-studies. RFSI outperformed the other methods with a RMSE in the mean daily temperature mapping case-study of 1.4 compared to RF and RFsp with RMSE 1.6, IDW with 1.8 and Space-Time Regression Kriging with 2.4. The covariants to model daily mean temperature were latitude, longitude, distance-to-coastline, elevation, seasonal fluctuation, insolation (total incoming solar radiation), and MODIS LST. They found that 'Seasonal fluctuation, MODIS LST images, insolation, and distance-to-coastline were the most important covariates for RF and RFsp', while in RFSI the observations from ground stations as covariants were the most important.

In [VBEM20] *Venter et al.* mapped T<sub>air</sub> over Oslo, Norway in 2018 on a scale of 10-30m using Sentinel, Landsat, LiDAR data which they combine with Netatmo PWS data from 1310 stations. They achieve an average RMSE of 0.52 °C ( $R^2 = 0.5$ ), 1.85 °C ( $R^2 = 0.05$ ) and 1.46 °C ( $R^2 = 0.33$ ) for annual mean, daily maximum, and minimum T<sub>air</sub>. The 'models performed best outside of summer months when the spatial variation in temperatures were low and wind velocities were high'.

Similarly, *Zumwald et al.* mapped hourly T<sub>air</sub> on a 10m<sup>2</sup> grid during a heat wave from 25-29th June 2021 in Zurich (Switzerland) and compared reference sensors, AWEL sensors

and PWS data from Netatmo stations and achieved RMSEs of  $1.43^{\circ}\text{C}$ ,  $1.35^{\circ}\text{C}$ , and  $1.71^{\circ}\text{C}$  for the respective types [ZKBK21]. They used QRF as their ML model of choice. Important features in their work were  $T_{air}$  at 2m (33%),  $T_{air}$  at 5cm (21%), relative humidity at 2m (13%), global radiation (8%), and more.

Both [ZKBK21, ZL12] found that there seems to be a systematic bias when estimating  $T_{air}$  using RF to underestimate high temperatures and overestimate low temperatures.

### 3.3.4 Histogram-Based Gradient Boosting

Similar to RFs, HistGB is an ensemble-based estimator that combines multiple estimators, in this case gradient boosting decision trees, and averages the results to get a more robust estimation. Compared to RFs, HistGB as implemented by sklearn<sup>2</sup> based on LightGBM [KMF<sup>+</sup>17], should have better performance, especially for bigger datasets and higher dimensional features, has built-in support for missing values which simplifies data pre-processing steps, and finally seems to slightly outperform RF according to the sklearn documentation<sup>3</sup> and other studies [AYDK22].

#### Bagging Methods

To decrease the variance of a single tree predictor, bagging is used to introduce randomization into the training process. There are several different bagging methods:

- **Pasting:** Splitting the data into different subsets and training a tree predictor on each subset [Bre99]
- **Bagging:** Splitting the data into different subsets but with replacement [Bre96]
- **Random Subspaces:** Splitting the data into different subsets of features [Ho98]
- **Random Patches:** Splitting both samples and features into different subsets [LG12]

RF and HistGB already use built-in bagging methods, so we do not consider these bagging methods further in this work.

### 3.3.5 Support Vector Machine Regression

SVMs are typically used in classification problems however they can be also used for regression problems, called Support Vector Regression (SVR). SVMs transform the input data into a higher dimensional space and try to find a hyperplane that separates the data into two classes. This approach is similar to linear regression; however, SVMs are

---

<sup>2</sup><https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingRegressor.html>, last accessed: 25.08.2023

<sup>3</sup>[https://scikit-learn.org/stable/auto\\_examples/ensemble/plot\\_forest\\_hist\\_grad\\_boosting\\_comparison.html](https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_hist_grad_boosting_comparison.html), last accessed: 01.09.2023

more robust to outliers and can be used for non-linear problems. Depending on the Kernel function used, e.g., Linear, Polynomial, Radial Basis Function (RBF), or Sigmoid, the model can suffer from the same problems as linear regression when dealing with correlated spatial data. As a result, appropriate counter measures need to be taken, such as using the Mahalanobis distance instead of the Euclidean distance for RBF kernels [KA06], which converts correlated features into uncorrelated features.

### 3.3.6 Neural Networks

ANNs are a more advanced ML method that takes inspiration from the human brain and electrical impulses being transmitted by neurons. An ANN is build-up of neurons which are grouped in layers that are connected and have activation functions and weights, that get trained during the learning process. The simplest ANN is the perceptron [Ros57] which models one single neuron, consisting of one or many inputs, a single processor, and a single output. An ANN usually consists out of one input and output layer, and one to many hidden layers. If there are more than one hidden layer, the ANN is usually called Deep Neural Network (DNN) and we speak from Deep Learning [LBH15]. ANNs have many hyperparameters and weights to train, therefore they need more data to train. Additionally, ANNs act more like a black box, so model analysis such as feature importance is not possible. Using ANNs is a trade-off between model capability, available data to train and test, and the explainability of the model.

As previously mentioned, the universal approximation theorem by Hornik et al. states how capable ANNs can be [HSW89]. Depending on how the model is setup, e.g., as a feed-forward network or a directed acyclic graph where layers feed back into each other, the type of loss-function, and the type of learning method, such as Stochastic Gradient Descend, ANNs can be adapted to many problems such as regression and interpolation. Due to the complexity of ANNs and the lacking explainability of the model, we do not focus on this ML method in this work however want to keep in mind that ANNs could be a powerful tool for tasks such as LST and  $T_{air}$  estimation and prediction [YSL<sup>+</sup>20].



## 4 Preparation of Datasets

Next to the ML model selected, the data used to train and evaluate the model has a major influence on its performance. If the data is not representative or information is missing about the underlying process that the model should be fitted to, there can be errors, bias, or an inability to generalize well to new data. In this chapter, we look at potential features for  $T_{air}$ , at data sources for these features, and discuss the construction process of the datasets used in this work.

In the field of natural language processing and computer vision, there is an abundance of large available datasets which have a big contribution to the advancements in the field, like annotated datasets as provided by Google Research<sup>1</sup>. In comparison in the field of climate research, there are also many datasets, including satellite data, weather station data, and climate model data; however, they are highly distributed or often not openly available. Platforms such as Google Earth Engine (Google EE) [GHD<sup>+</sup>17] try to address this issue; however, the dataset catalogue<sup>2</sup> is still limited and does not include datasets offered by local authorities or other research institutions, such as universities.

For the specific use case of temperature interpolation in urban areas, an optimal dataset would contain high spatial and temporal resolution sensor data, e.g., a high sensor density and a low time interval of for example five to ten minutes. Additionally, the sensor placement and sensor quality have a high influence on the accuracy of the sensor readings [Oke06], therefore the correct placement and calibration of the sensors needs to be guaranteed. Such requirements are not met by traditional weather station networks as the spatial coverage is too low, as a single weather station is not enough to capture the urban microclimate [OMCV17]. The weather station locations of the DWD are shown in figure 4.7, which shows that usually at most one weather station is available per city. The weather stations however offer a high temporal resolution in addition to very high-quality sensors, which is why they can be used as reference stations for quality control of other sensors, as discussed in section 4.4.

A solution to the problem of low spatial coverage is the usage of sensor networks. There are several projects that run dense urban-climate monitoring networks [MCG<sup>+</sup>13] such as the Helsinki Testbed [KPS<sup>+</sup>11] or the Birmingham Urban Climate Laboratory [WYC<sup>+</sup>16]; however, access to those datasets is limited, for example due to outdated links or the need to request access. Due to the high cost of running such dense sensor networks with high sensor and maintenance costs, professionally run sensor networks are rare and are

---

<sup>1</sup><https://research.google/resources/datasets/>, last accessed: 05.08.2023

<sup>2</sup><https://developers.google.com/earth-engine/datasets>, last accessed: 05.08.2023

often only run for a limited period until project funds run out.

As an alternative, sensor networks can also be crowdsourced and run by citizens, distributing the cost of individual sensors as well as maintenance costs among many. Especially with advances in sensor technologies, lost cost and compact sensors are more affordable than ever while still providing good data quality [Gri06, RGA<sup>+</sup>09]. In the context of meteorological data, such sensor networks are often referred to as PWS [MFG<sup>+</sup>17] or PWS [HGMS<sup>+</sup>22] networks. In this work, we use the term PWS.

The main downside of this approach is the lack of quality control and meta data, as the sensors are usually placed by non-professionals in suboptimal locations, e.g., in direct sunlight or too close to walls, leading to incorrect readings or bias in the data. A lack in meta data can also lead to issues, if for example exact positions of the sensors are not known and information about the height of the sensor above ground is missing. However, other concerns such as data privacy also need to be accounted for, as such weather stations are often placed on private property.

In this work, data from PWS networks is used to create datasets for the training and evaluation of ML models for  $T_{air}$  interpolation. In the following sections, we look at available PWS providers and their data and potential features for  $T_{air}$  interpolation, discuss additional pre-processing steps such as quality control or sensor height correction, and finally discuss the construction of the datasets used in this work.

## 4.1 Private Weather Station Network Providers

PWS providers offer a platform for users to upload their sensor data and either sell weather stations and sensors themselves (e.g., Netatmo) or provide guides to allow users to connect their own sensors to the platform (e.g., Sensor.Community). Netatmo data in particular has been used in several studies [MFG<sup>+</sup>17, HGMS<sup>+</sup>22, VBEM20, ZKBK21] and has seen complementary studies for example discussing QC

processes [FBD<sup>+</sup>21], later seen in Section 4.4. There are also other PWS network providers such as Sensor.Community or WOW that have been used in several studies [HKS<sup>+</sup>14]. To find out which provider best fits our needs in this work, the following section compares the different providers and their data.

### 4.1.1 Sensor.Community

Sensor Community<sup>3</sup> is a contributors driven global sensor community that creates Open Environmental Data and has an archive<sup>4</sup> of their historical sensor data world-wide. There are no quality measures recorded for each sensor, but as crowd-sourced sensor data tends to have a lower quality than professionally setup sensors, e.g., sensor placement by non-professionals, we need to explore how the data quality looks like. In Figure 4.1,

---

<sup>3</sup><https://sensor.community/en/>, last accessed: 05.08.2023

<sup>4</sup><https://archive.sensor.community/>, last accessed: 05.08.2023

where we see the greater Hamburg area with a currently reported temperature of 25°C by the DWD Fuhlsbüttel station, there are multiple sensors that report a temperature of 30°C and above, which could be either due to the sensor being placed in direct sunlight or due to the sensor being faulty. An outlier near Pinneberg is shown in Figure 4.2, where one sensor reports 25°C, as currently expected, and one sensor reporting 50°C, which is clearly an outlier. This data quality issue needs to be addressed in the data pre-processing step and can result in a significant reduction of available data. This was also an issue discussed in [MFG<sup>+</sup>17], as “erroneous metadata, failure of data collection, and unsuitable exposure of sensors lead to a reduction of data availability by 53 %”. From a meta-data perspective, there is no information on the sensor height above ground as well as no quality measures for each sensor, or information on the sensor location accuracy.

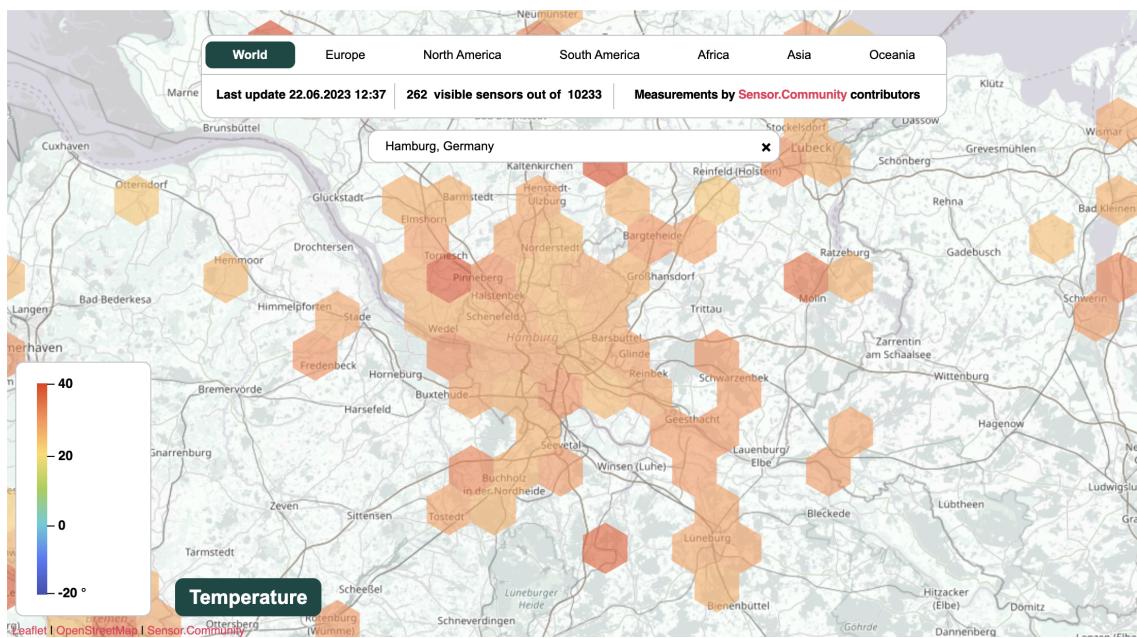


Figure 4.1: Temperature map from Sensor Community for Hamburg, Germany, on 22.06.2023 12:51h with the DWD reference at 25°C

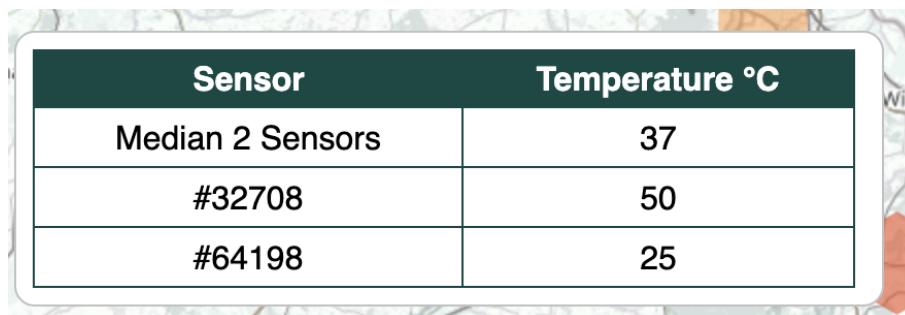


Figure 4.2: Temperature outlier from Sensor Community for Hamburg, Germany, on 22.06.2023 12:51h with the DWD reference at 25°C

Overall, there are around 11.738 active sensors<sup>5</sup>. Of these sensors, many are located in

<sup>5</sup>as of 24.06.2023

Germany, as seen in Appendix 4, and almost half of them are of type BME 280, which is a low-cost Bosch sensor which can measure temperature, pressure, and humidity. The sensor locations as of May 2023 are shown in 4.3. DHT22 sensors can measure temperature and humidity, BMP280 and BMP180 sensors can measure temperature and pressure, and BME280 sensors can measure temperature, pressure, and humidity.

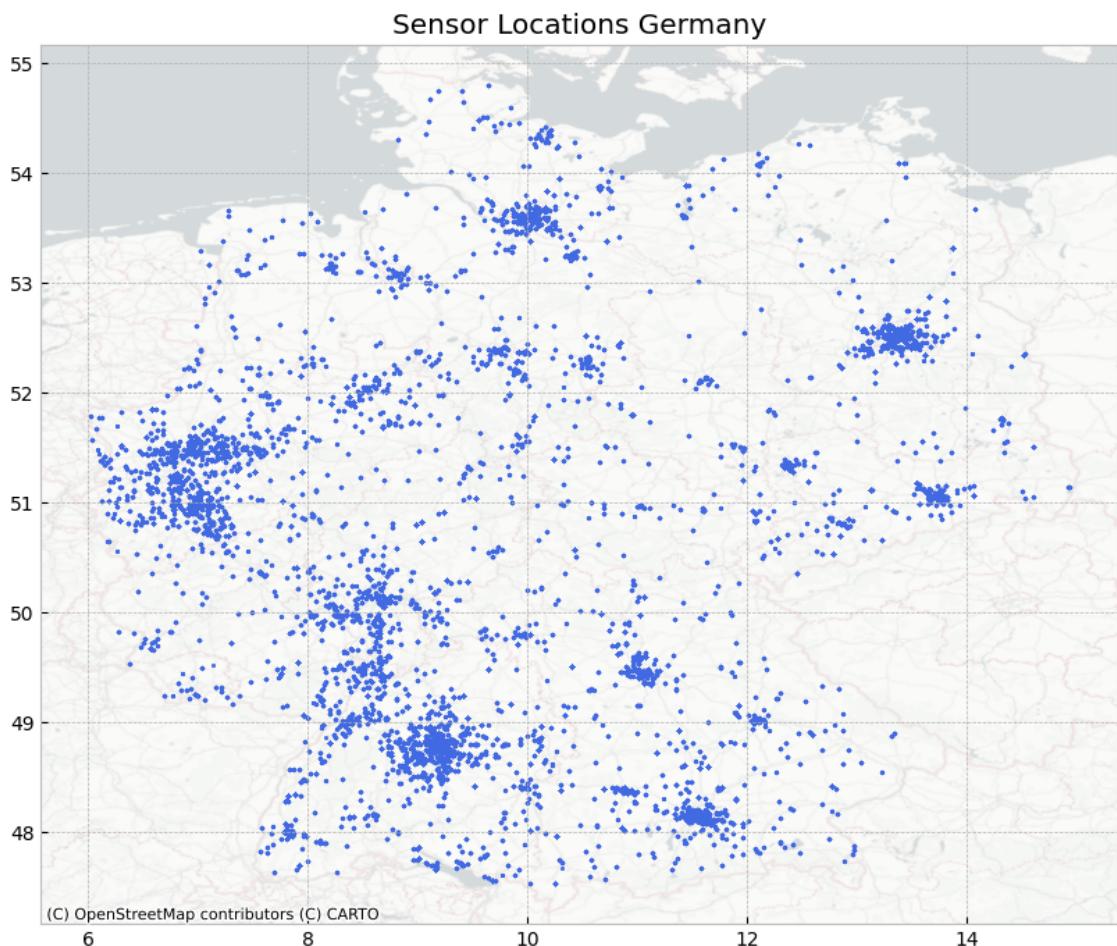


Figure 4.3: Sensor locations of Sensor Community in Germany, as of 01.05.2023, of sensor type DHT22 (2590 sensors), BME280 (1558 sensors), BMP280 (100 sensors), BMP180 (72 sensors)

#### 4.1.2 Netatmo

Netatmo<sup>6</sup> is a French company that sells smart-home devices including outdoor weather stations, indoor sensors for air quality, as well as other products such as smart cameras. They host a weather map<sup>7</sup> where customers can share their outdoor weather station data. They provide an API to access current weather station data as well historic data from in-

<sup>6</sup><https://www.netatmo.com/en-eu>

<sup>7</sup><https://weathermap.netatmo.com/>, last accessed: 06.08.2023

Measurement	Unit	Measurement Range	Precision	Recording Frequency
Temperature	°C	-40°C to 65°C	0.3°C	averaged over 5 min
Humidity	% (RH)	0 to 100%	3%	-
Air Pressure	mbar	260 to 1160 mbar	1mbar	-
Noise	dB	35 to 120 dB	-	-
Wind Speed	m/s	0 to 45 m/s (160 km/h)	0.5 m/s	every 6 sec, averaged over 5 min
Wind Direction	°	0 to 359°	5°	every 6 sec, averaged over 5 min
Rainfall	mm/h	0.2 to 150 mm/h	1mm/h	every 5 min (bucket is emptied)

Table 4.1: Netatmo Sensor Specifications (Vendor reported)

dividual outdoor sensors and modules. They provide their historical and current weather data for commercial partners or partners in the research and education sector. They are part of the EUMETNET project <sup>8</sup> which is a network of 31 European meteorological and hydrological services. The project aims to facilitate the exchange of weather data and to improve the quality of weather forecasts, especially in the context of PWS [HGMS<sup>+</sup>22]. There are currently no openly historical datasets available from Netatmo data, only private datasets <sup>9</sup> that are only available for partners such as EUMETNET members. They offer an educational program <sup>10</sup> to access temporally and spatially limited amounts of data that is usually only available to commercial partners.

In the context of collecting meteorological data, the smart weather products are of particular interest. These include a smart outdoor weather station that collects  $T_{air}$ , humidity and air pressure, an anemometer that collects wind speed and direction, and a rain gauge. The sensor specifications, as reported by the vendor himself, is reported in Table 4.1.

In this work, data from Netatmo stations is used as Netatmo offers many sensors in Germany in urban areas, exemplified by Figure 4.4 for the region of Hamburg, and by Figure 4.5 for the region of Stuttgart. The developer portal <sup>11</sup> offers a way to programmatically access all public sensor measurements via a REST API; however, each request has a limit on the spatial extend of the requested area for the current weather data. For historic data, the limit per request per sensor is 1024 data points. The API has a tight rate limit per application. For applications below 100 users, the rate limit is 2000 requests every hour and 200 requests every 10 seconds across all users, and 500 requests every hour and 50 requests every 10 seconds per user <sup>12</sup>. In this work, we use the REST API to collect sensor data from Netatmo sensors.

<sup>8</sup><https://www.eumetnet.eu/>, last accessed: 06.08.2023

<sup>9</sup><https://catalogue.ceda.ac.uk/uuid/e8793d74a651426692faa100e3b2acd3>, last accessed: 06.08.2023

<sup>10</sup><https://www.netatmo.com/en-eu/weather-with-netatmo>, last accessed: 01.09.2023

<sup>11</sup><https://dev.netatmo.com/apidocumentation>, last accessed: 01.09.2023

<sup>12</sup><https://dev.netatmo.com/guideline#rate-limits>, last accessed: 06.08.2023

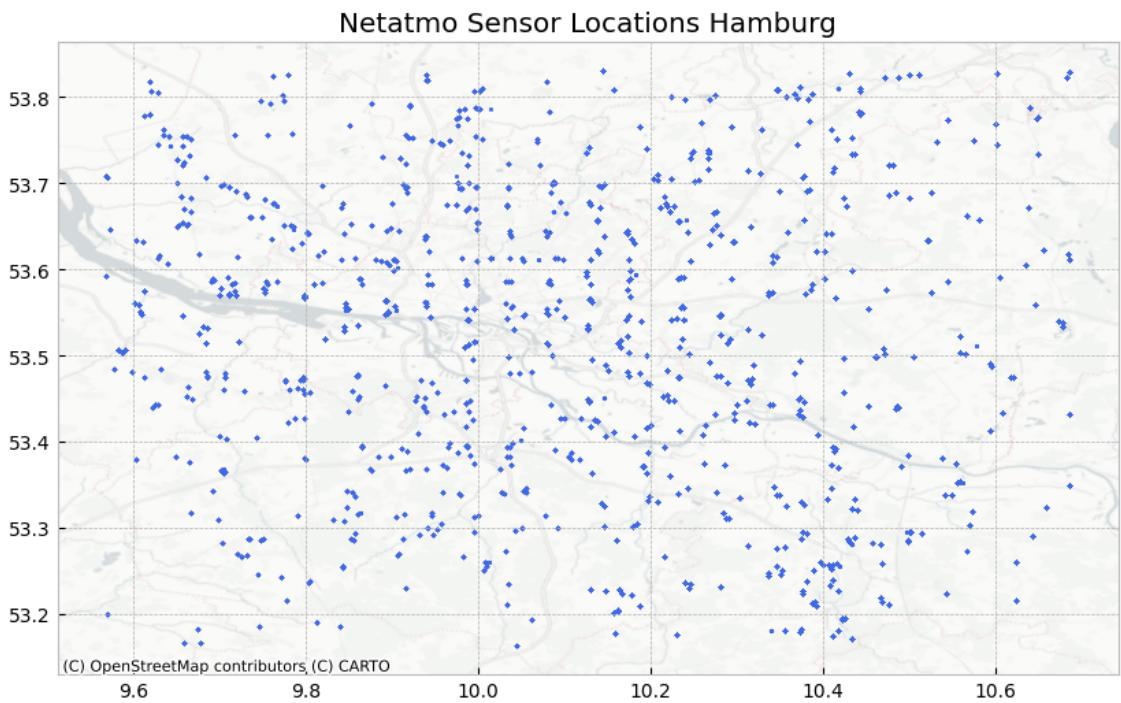


Figure 4.4: Sensor locations of Netatmo in Hamburg, Germany, as of 28.06.2023

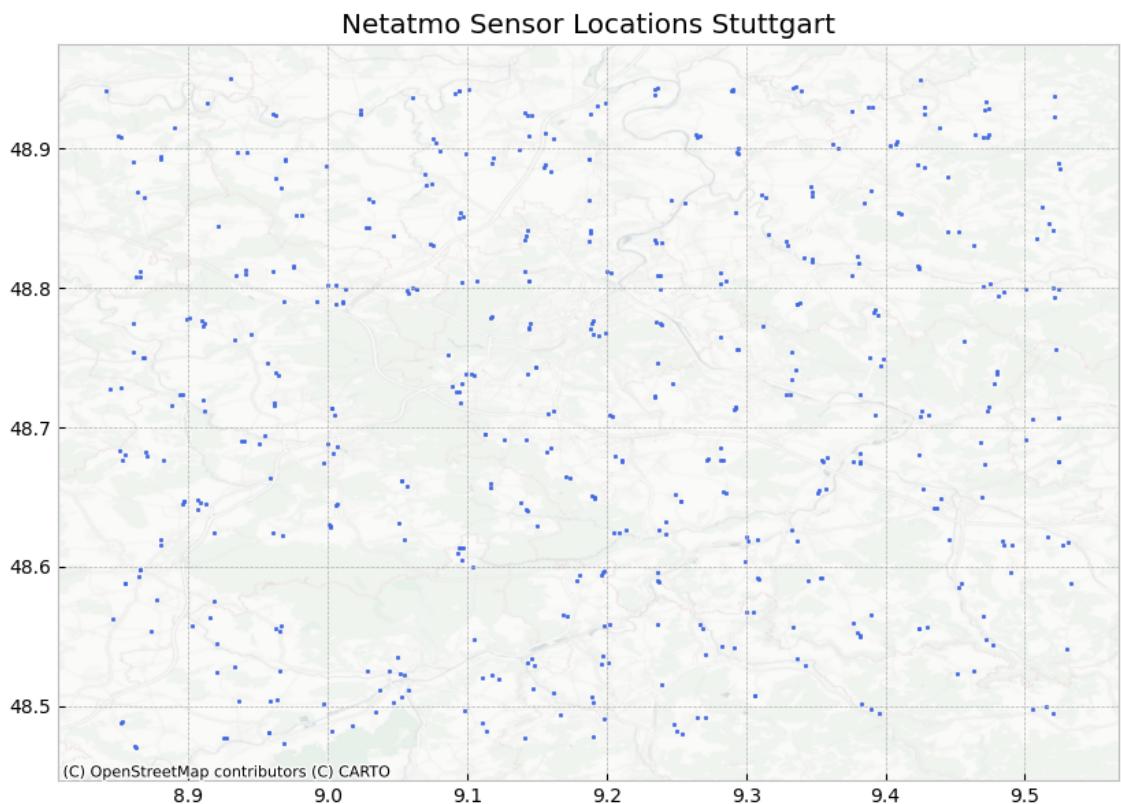


Figure 4.5: Sensor locations of Netatmo in Stuttgart, Germany, as of 19.06.2023

---

## Quality Considerations

Netatmo sensors have a good measurement accuracy; however, due to the compact design, an aluminium housing, poor ventilation due to the small case, no dedicated radiation screen resulting in a proneness to radiative errors, and therefore overall slow sensor-response time [MFG<sup>+</sup>17, Büc18], Netatmo weather stations have a systematic bias that influences data quality. Due to the uniformity of Netatmo sensors, e.g., all sensors are built in the same way, this bias could be corrected in the QC step; however, this is not further explored in this work.

### 4.1.3 Other Providers

Other sources for crowdsourced weather station data include WOW<sup>13</sup> and Weather Underground<sup>14</sup>. WOW is a platform run by the UK Met Office, which is the UK's national weather service, and has a dense sensor coverage in the UK and the Netherlands as seen in figure 4.6. Weather Underground is a commercial weather service which also provides a crowdsourced weather station network. Unfortunately, Weather Underground only provides an API for users with a registered weather station or other bulk download options for historical data. The website would allow for manual download of historical data, but this is not feasible for the amount of data needed in this work.

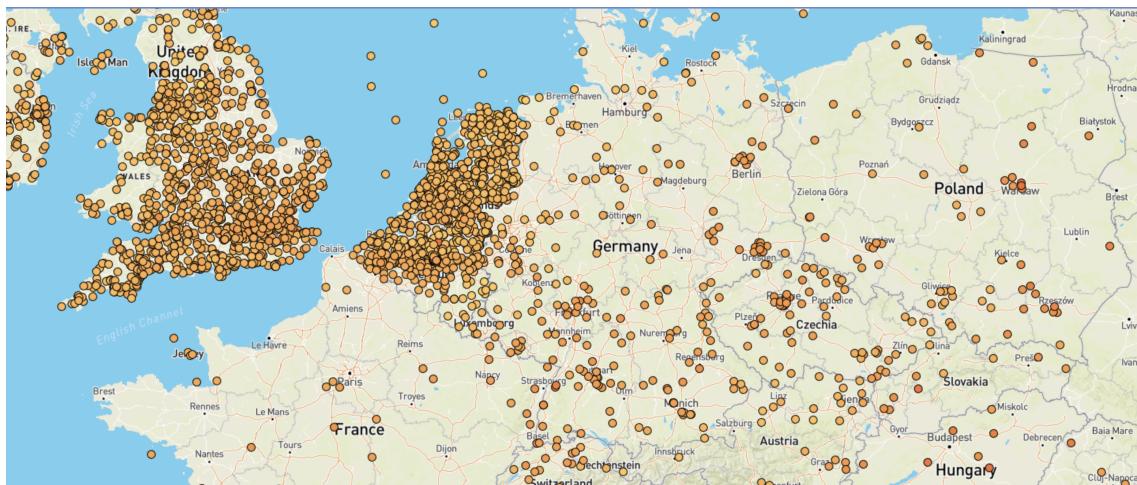


Figure 4.6: Temperature sensor locations from WOW, accessed on 05.07.2023

## 4.2 Reference Data Providers

In order to add additional validation to crowdsourced weather data, reference data from (official) weather stations can be used. These weather stations should be setup according to current WMO guidelines [WMO18] to ensure high data quality. These standards are

<sup>13</sup><https://wow.metoffice.gov.uk/>, last accessed: 01.09.2023

<sup>14</sup><https://www.wunderground.com/>, last accessed: 01.09.2023

either achieved by official weather services, or by institutions such as universities whose sensors are maintained by experts.

#### 4.2.1 DWD

The DWD has many objectives that are defined by the DWD-law in Germany. Its tasks include meteorological and climatological monitoring of the atmosphere, meteorologically securing the airspace for civil aviation, monitoring the Maritim climate, and more. The DWD operates a large monitoring network and publishes most of its data via its Open-Data portal<sup>15</sup>.

The main advantages of the DWD data are high data quality through reference instruments and proper setup according to WMO guidelines [WMO18]. The main disadvantage is the low spatial coverage of the data, as stations are sparsely distributed to measure the overall mesoscale climate, as seen in Figure 4.7. Additionally, a lot of the public weather stations are located close to airports, which are usually located outside of cities, and therefore not suitable for measuring urban microclimates.

#### Urban Climate Stations

Next to the official weather stations, the DWD also operates urban weather stations; however, there are currently only four stations in the following cities:

- Berlin-Alexanderplatz, Berlin, Berlin
- Freiburg-Mitte, Freiburg, Baden-Württemberg
- Hannover-Nordstadt, Hannover, Niedersachsen
- Dresden-Neustadt, Dresden, Sachsen

The number of urban weather stations is planned to be gradually extended to reach 10 stations with the locations being primarily determined by the measurement objectives such as determining a city's maximum UHII<sup>16</sup>. Due to the low number of weather stations, their data is not used in this work.

#### Weather Radar

The DWD also operates a network of weather radars<sup>17</sup> that are used to measure precipitation and wind speed. This data could be interesting in the context of air interpolation in order to detect precipitation events that have a major influence on humidity and temperature or to detect wind speeds that also play an important factor in dissipating heat

<sup>15</sup><https://opendata.dwd.de/>, last accessed 13.07.2023

<sup>16</sup>[https://www.dwd.de/EN/climate\\_environment/climateresearch/climate\\_impact/urbanism/urban\\_heat\\_island/urbanheatisland\\_node.html](https://www.dwd.de/EN/climate_environment/climateresearch/climate_impact/urbanism/urban_heat_island/urbanheatisland_node.html), last accessed: 01.09.2023

<sup>17</sup><https://www.dwd.de/DE/leistungen/radarprodukte/radarprodukte.html>, last accessed 12.07.2023, not available in english

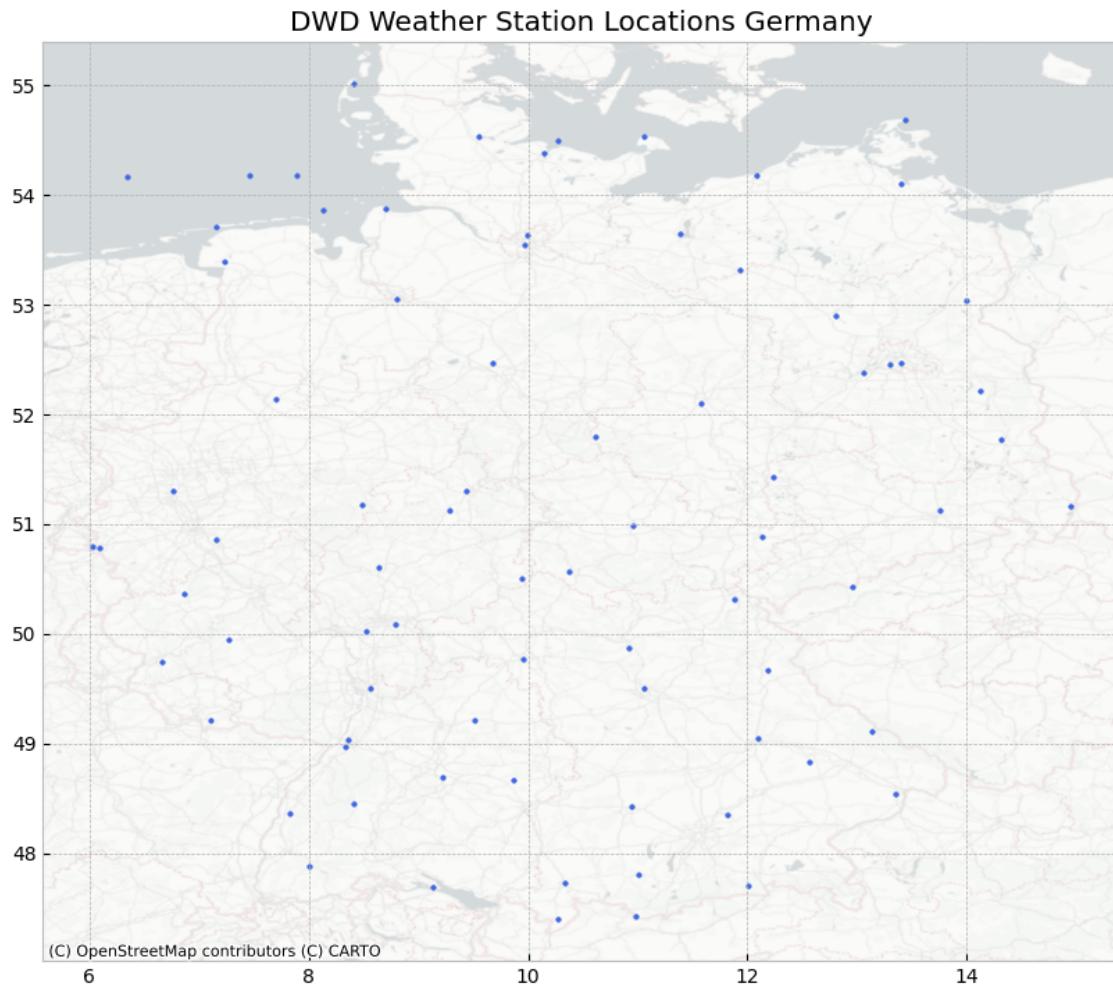


Figure 4.7: DWD Weather Station Locations in Germany, [https://opendata.dwd.de/climate\\_environment/CDC/observations\\_germany/climate/subdaily/standard\\_format/KL\\_Standardformat Beschreibung\\_Stationen.txt](https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/subdaily/standard_format/KL_Standardformat Beschreibung_Stationen.txt), accessed 28.06.2023

and transporting it away from urban areas. Due to the limited scope of this work, this data is currently not used.

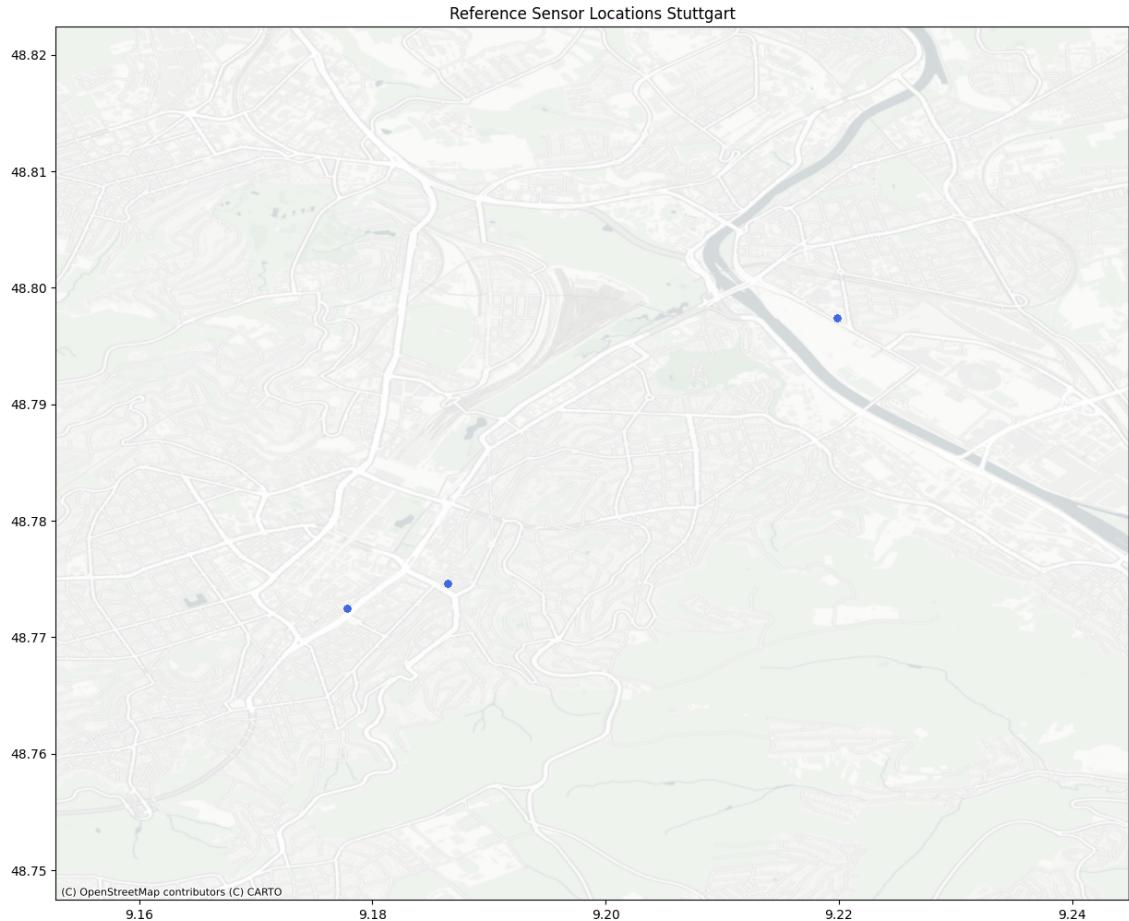
## 4.2.2 Locally Operated Weather Stations

Next to official weather services, many public and private institutions operate weather stations. The following section lists two examples of such institutions, the Meteorological Institute of the University of Hamburg and the Office for Environmental Protection of the City of Stuttgart.

### University of Hamburg – Meteorological Institute

In Hamburg the Meteorological Institute of the University of Hamburg addresses local climate concerns, supports many projects, and support decision makers relating climatic

concerns. One of its projects is the Hamburg Urban Soil Climate Observatory (HUSCO) which operates multiple urban climate weather stations that measure  $T_{air}$  at 2m height as well as many other parameters including soil temperatures and conditions. In the current scope of this work we do not use data from this project; however, it could be a good reference point especially in the context of 2m  $T_{air}$  for CUHI detection and wanted to mention it here.



**Figure 4.8: Weather Station Locations in Stuttgart,** <https://www.stadtklima-stuttgart.de/>, last accessed: 10.08.2023

### City of Stuttgart – Office for Environmental Protection

The Office for Environmental Protection of the city of Stuttgart has the goal of monitoring the climate in Stuttgart and the surrounding area and improve the living conditions of the citizens. The main focus of the office is on air quality, noise, and (urban) climate. It operates several weather stations in the city of Stuttgart, which are shown in Figure 4.8. The data from these stations is published directly on the website, a detailed quality assessment is given, and reference grade sensors are used. Unfortunately, the stations are mounted on top of building, e.g., 20-23m above ground, which is not ideal for measuring urban microclimates as the  $T_{air}$  is several degrees cooler than directly on the

ground. This example underlines the importance of proper sensor placement and metadata documentation. The data from the weather stations could be corrected using the DWD reference station nearby; however, uncorrected, the  $T_{air}$  cannot be used to validate the crowdsourced data. Due to the high placement, other measurements such as wind direction and speed could be used to get a good estimation of the overall climate in the city; however, not on the microscale.

## 4.3 Remote Sensing Data Providers

### Remote Sensing

In comparison to stationary sensors that are installed directly in the environment they are observing, remote sensing describes the process of observing a target environment from afar [CW11]. In climatology, remote sensing is used to collect meteorological data via satellites, planes, or balloons by either capturing image data that can be used to identify things like cloud and land coverage, by measuring passive radiation, or by actively sending out microwaves or using LiDAR to detect features such as LST. Remote sensing comes with its own set of advantages and challenges.

The major upside of sensors moving way above ground is the high spatial coverage, that allows for meso- and planetary-scale analysis of weather phenomena. Another upside is the great data availability, as many satellite providers (e.g., NASA, ESA, etc.) publish their satellite data. This creates many research opportunities and services directly relying on these measurements.

Remote sensing also comes with certain downsides. The primary downside is the low spatio-temporal resolutions. Weather satellites usually are not orbit-stationary and move around earth on a predetermined orbit. Consequently, satellites only pass over each individual area a couple times a day, making real-time applications for currently unobserved areas impossible. Additionally, the spatial resolution can be too low for micro-/local-scale analysis, with typical LST resolution spanning from  $1 \text{ km}^2$  to tens or even hundreds of  $\text{km}^2$  per data point/grid field. In the atmosphere, there is also a lot of environmental noise, like radiation, that can have a negative influence on the measurement accuracy. Another disturbing factor can be clouds or other types of particles like rain that absorb radiation/microwaves sent from the sensors, making measuring under cloudy/rainy conditions either impossible. There exist methods to estimate values instead, relying on outgoing radiation from the surface; however, these approaches are usually less accurate. These restrictions highly depend on the sensor used, as different sensors use different technologies, e.g., microwaves with different wave lengths or higher resolution sensors.

### 4.3.1 Google Earth Engine

Google EE is a science data and analysis platform that allows users to work with and transform massive datasets with remote sensing data that are available as OpenSource [GHD<sup>+</sup>17]. Remote sensing data in this work is processed and obtained via this platform and used datasets are cited. Datasets available on Google EE are not published by Google itself but from institutions such as NASA, ESA, or the European Union.

## 4.4 Quality Control

QC is an essential step in the process of data analysis and preparation. The goal is to identify and remove outliers in the data that are due to placement errors of sensors, sensor malfunctions, sensor inaccuracies or other errors. In the context of PWS, weather stations are placed and maintained by non-professionals, making QC even more important. One of the main challenges in the context of (hyper-) local urban  $T_{air}$  data is to not flag data as outliers that is representative of the local climate in case of extreme temperature, e.g., heat islands, and at the same time identify erroneous or wrongly placed sensors, e.g., too close to walls, in direct sunlight, indoors, etc. Additionally, current PWS networks do not track sufficient metadata on the sensor placement, e.g., sensor height, which also plays an important role in protecting the privacy of citizens and not exposing too accurate sensor locations.

Due to the popularity of Netatmo weather station data in research due to high spatio-temporal resolution, there are several software libraries available that help simplify and automate the QC process. These tools were primarily developed for Netatmo temperature data; however, CrowdQC and TITAN can also be used for other nearly-normally distributed data sources [HGMS<sup>+</sup>22]. The following tools are available:

- CrowdQC (R package <sup>18</sup>)
- CrowdQC+ [FBD<sup>+</sup>21] (R package <sup>19</sup>)
- TITAN (R package <sup>20</sup>)
- NetatmoQC (Python 3 package <sup>21</sup>)

In this work, CrowdQC+ is used for QC as it offers improvements and bug fixes compared to CrowdQC. It's an open-source software library written in R, a popular programming language for statistical applications. The data needs to be in the following format:

- $p\_id$ : The unique ID of the station

---

<sup>18</sup><https://doi.org/10.14279/depositonce-6740.3>

<sup>19</sup><https://github.com/dafenner/CrowdQCplus>, last accessed: 01.09.2023

<sup>20</sup><https://github.com/mtno/TITAN>, last accessed: 01.09.2023

<sup>21</sup><https://source.coderefinery.org/iOBS/wp2/task-2-3/netatmoqc>, last accessed: 01.09.2023

---

- *time*: The time of the measurement
- *ta*: The  $T_{air}$  in degree Celsius
- *lon*: The longitude of the station
- *lat*: The latitude of the station
- *z*: The height of the station in meters, optional

The CrowdQC+ library implements the following required steps of QC: Metadata Check, Distribution Check, Data Validity, Temporal Correlation, Spatial Buddy Check. There are also the following optional steps available, that are currently not used: Temporal Interpolation, Daily Validity, Validity in Time Period, and Correction for Time Constant. The steps used in this work are shown and explained in Table 4.4.1, including the number of data and stations available after each step.

In their own study, CrowdQC+ kept 47.1% and 69.2% of data after steps m1-5, and only 20.7% and 29.5% after steps o1-o3, for the cities Amsterdam and Toulouse respectively [FBD<sup>+</sup>21], given default parameters. In that setting, CrowdQC kept more data with 41.0% in Amsterdam and 54.9% in Toulouse. In this work, CrowdQC+ is used with default parameters excluding height validation, as this data was not available for almost all sensors. Additionally, only the first 5 required steps are used, as the optional steps are not needed for the interpolation. The input data for CrowdQC+ also needs have the same temporal resolution and intervals.

In this study, we use a 10 min interval to have a high temporal resolution and use the default parameters except excluding the height check due to the missing values. Important to note here, that in the following, only the  $T_{air}$  is validated and no other measurements such as pressure or humidity. CrowdQC+ could be used for other approximately normally distributed features; however, there hasn't been more specific research in this direction. We assume that a station that seems to be setup correctly and produces good  $T_{air}$  measurements, also captures the other measurements correctly for simplicity reasons.

#### 4.4.1 Quality Control for Sensor.Community

For Sensor.Community, we can see several interesting things for the  $T_{air}$ . The first is, that in January 2023 less data is lost due to QC compared to June 2023. This could be to the fact, that in colder environments with less solar radiation, sensor placement, for example close to buildings, has less of an influence. In comparison, June 2023 had many hotter days, therefore it could be that more extreme readings are flagged as outliers. We can also see that Sensor.Community loses a lot of stations in step m4 in June 2023 compared to January 2023, which is the temporal correlation with the median of all stations. This could also be due to a higher temperature difference across Germany, therefore comparing smaller areas could be beneficial for this. We can also see, that the m5 check, which

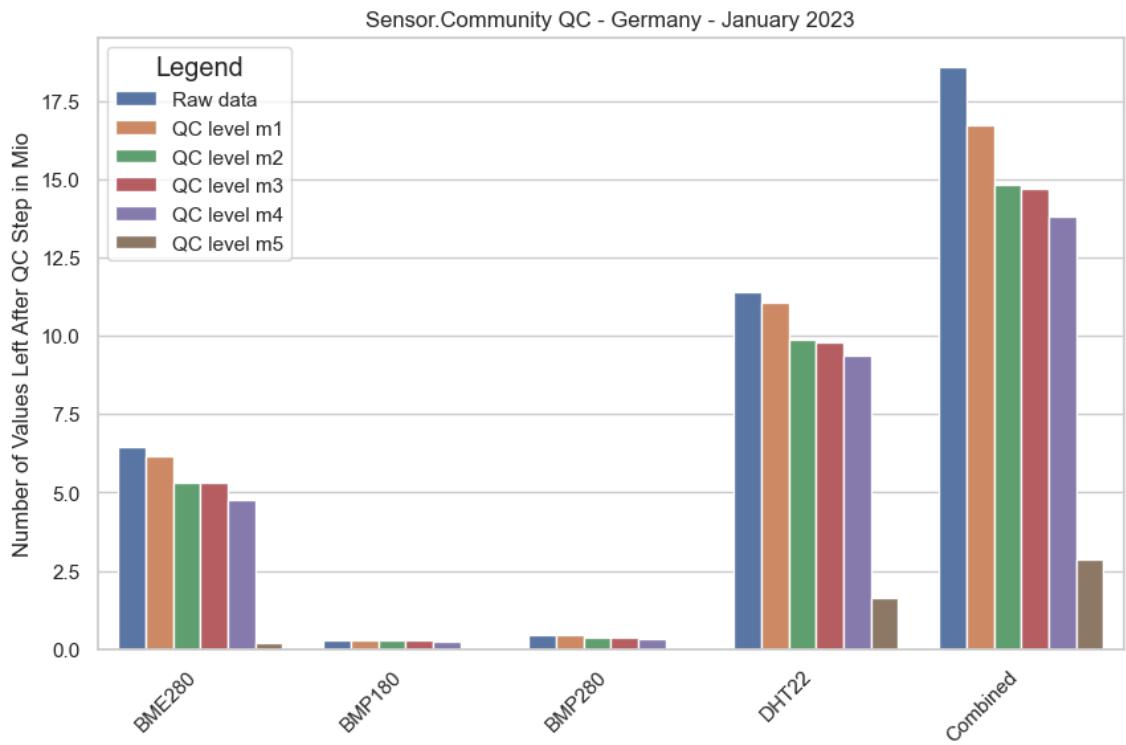


Figure 4.9: QC Results for Sensor.Community Data for Germany, January 2023

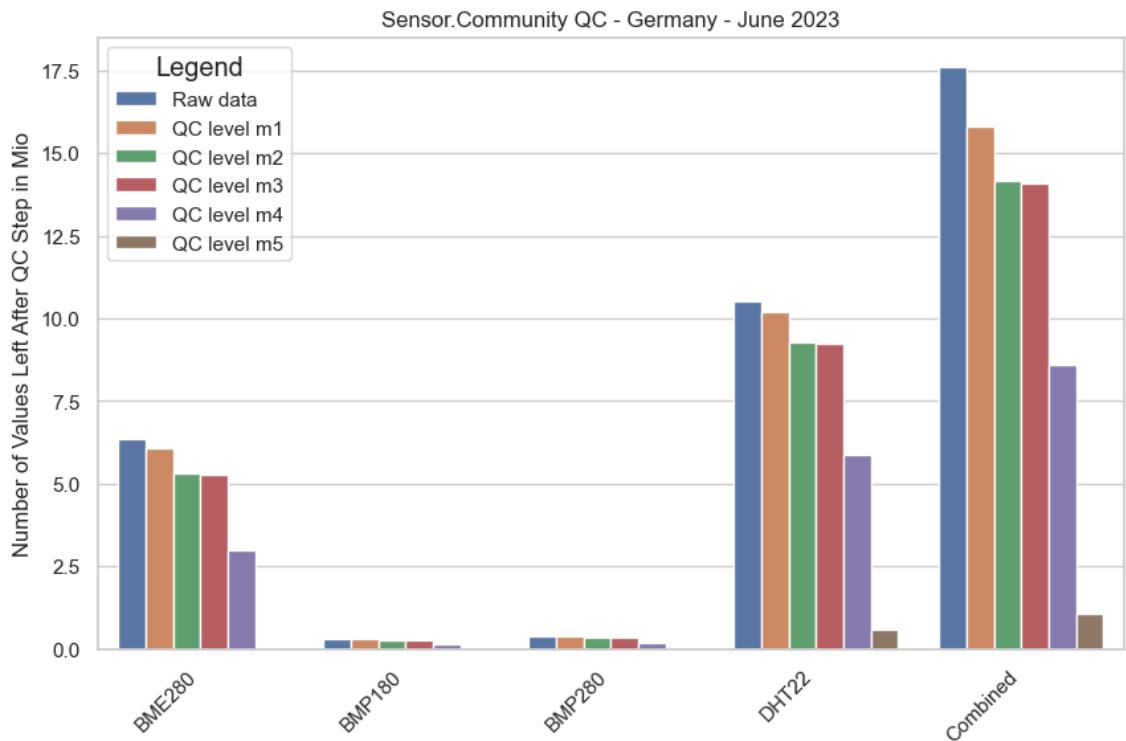


Figure 4.10: QC Results for Sensor.Community Data for Germany, June 2023

is the buddy check with surrounding stations, also removes a lot of stations which could be due to the low station density.

The CrowdQC+ library can theoretically be used to validate other near-normally distributed variables; however, this could change the way the QC step parameters should be set and changed from the default. For the  $T_{air}$ , the default parameters as proposed by the library were used. Due to the limited scope, for other readings, e.g., relative humidity and atmospheric pressure, we simply remove default values. As an improvement, for other variables a more sophisticated QC process should be used. In addition, due to the low sensor density and the fact, that all types of sensors used are good LCSs, we simply combine all sensor readings after QC step m5 into one dataset and ignore the sensor type. After the QC process, the Sensor.Community sensor locations left are shown in 4.11. In this figure we can see, that there are many sensors left in Stuttgart, Hamburg, Munich, and some in Cologne. Due to DWD stations only being present in Hamburg and Stuttgart, those two areas are candidates for further usage.

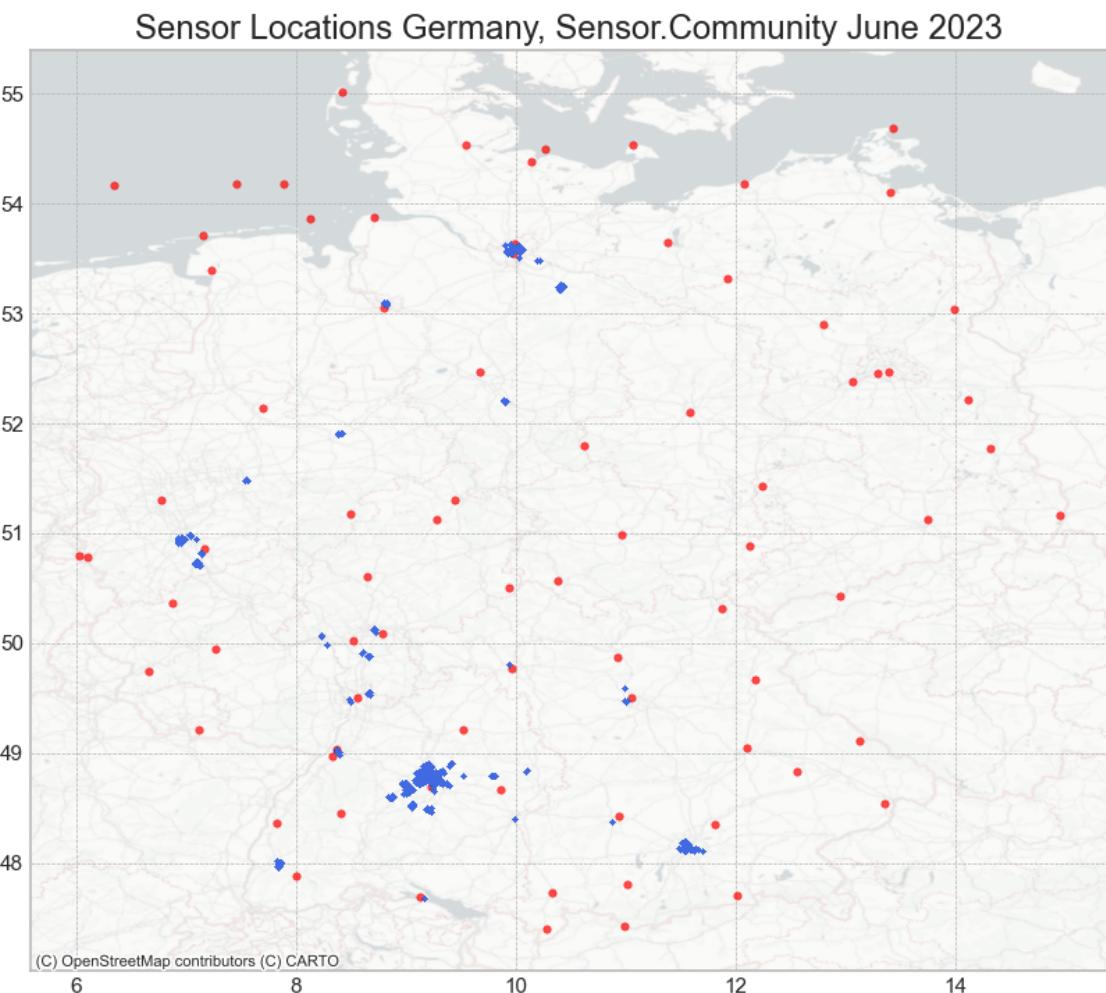


Figure 4.11: QC Results for Sensor.Community for Germany, June 2023

Table 4.2: Quality Control Steps of CrowdQC+

Required	<b>Id</b>	<b>Name of Step</b>	<b>Functionality</b>	<b>% of Data</b>	<b>Num Stations</b>	<b>Num Values</b>
	m1	Metadata Check	Validates longitude and latitude values and removes stations with identical values. Mainly aims to remove stations with default values from locations from IP addresses due to improper configuration by the end-user	97.40%	1077	2.082.283
	m2	Distribution Check	Primarily targets radiative error that lead to unrealistic high ta values and sensors installed indoors	86.50%	1041	1.849.247
	m3	Data Validity	Checks values of stations that did not pass m2. If more than 20% of data didn't pass the check, the station is considered to be faulty and is removed	85.20%	845	1.821.479
	m4	Temporal Correlation	Checks the temporal correlation between each station and the median of all stations for a specified period of time, default 1 month. Targets indoor stations that have weak temporal correlation to the median of all stations.	79.90%	829	1.708.061
	m5	Spatial Buddy Check	Neighbourhood-based check to identify outliers within a specific area. Primarily targets radiation errors with too high ta values. Defaults to radius of 3000m and 5 neighbours.	31.53%	466	674.004
Optional	o1	Temporal Interpolation	Step to interpolate missing values in the time-series of each station to increase data availability	-	-	-
	o2	Daily Validity	Verifies robust calculations of daily values	-	-	-
	o3	Validity in Time Period	Checks if enough values are available in a given time frame Sensors have different times that they respond to ta changes.	-	-	-
	o4	Correction for Time Constant	Due to Netatmo design flaws, a constant correction for all stations can be applied.	-	-	-

This table shows the QC steps used in this work from the CrowdQC+ library, including the % of data available after each step, the number of stations available and the number of values left after each step. Optional steps are currently not used.

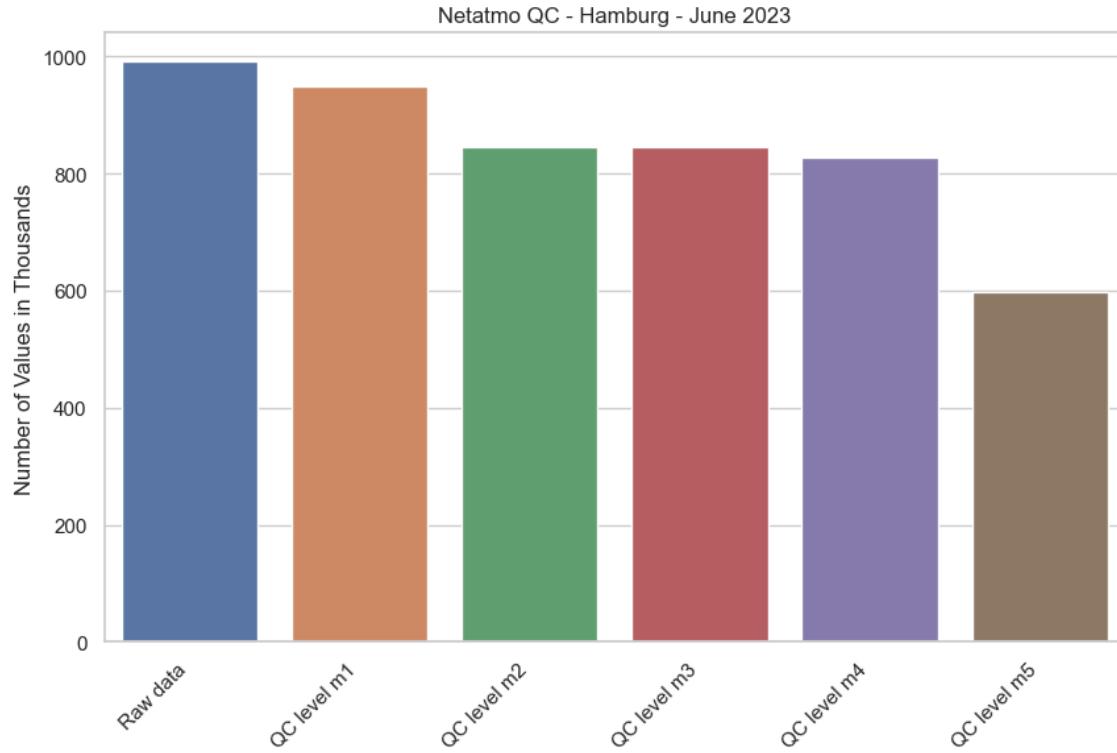


Figure 4.12: QC Result Statistics for Netatmo Data for Hamburg, June 2023

#### 4.4.2 Quality Control for Netatmo

The QC results for Netatmo stations in Hamburg during June 2023 can be found in 4.12 which shows the absolute values of sensor readings available after each step including the overall available data as ‘Raw data’ before QC which is 1.026.721 rows. After QC 60.24% of Netatmo data is still available in Hamburg (so a loss of 39.76%). The amount of data kept is in line with comparable cities as tested by CrowdQC+’s study which kept 47.1% and 69.2% for the cities Amsterdam and Toulouse respectively after steps m1-m5 [FBD<sup>+</sup>21] as mentioned above. Another comparison can be made to the study by Meier et al. which kept 47% of Netatmo data in Berlin after QC with CrowdQC [MFG<sup>+</sup>17]. It is interesting to note that stations directly next to water seem to be removed proportionally more often. This could be due to higher variability of wind close to water [HKS<sup>+</sup>14] which can result in higher prediction errors.

A station is not necessarily removed at all times, only if it shows outliers for a certain period after QC step m3, maybe due to improper setup. The difference between day and night for an example day of 19.06.2023 can be found in Appendix 1 Figure 1.

## 4.5 Feature Engineering

The goal of feature engineering is to create features from the available data that can be used as input for the machine learning models. Based on the features, different models

can be used for completely different tasks such as interpolation vs. extrapolation. The process includes the selection of features, the extraction of features from the raw data, and the transformation of features into a format that can be used by the machine learning models which could include scaling and/or normalization. Especially in the context of  $T_{air}$  interpolation, a lot of domain knowledge is required to select the right features and model correlations between them correctly. The target feature in this work is the  $T_{air}$  at canopy height, e.g., 2m height so that CUHIs could be detected when using these trained interpolation models. Alternatively, the target temperature could also be adapted to be measured at another height or be exchanged based on the use-case.

The input features are a combination of sensor readings from PWS networks such as Sensor.Community and Netatmo, satellite data such as land cover and vegetation health or LST, and additional meta data such as soil conditions, zoning plans, or locations of sensors. The goal of this section is to give an overview of the different features that can be used for  $T_{air}$  interpolation and discuss several highly important features that are especially relevant in the context of urban microclimate based on related studies. Important questions are: 1. What features are there and how can they be measured/sourced from? 2. How do features for the 2 use cases, e.g., single station interpolation vs. areal interpolation, differ? 3. How do we handle QC outputs? 4. How do we deal with correlations between inputs so that we do not introduce bias into our models?

#### **4.5.1 Feature Engineering Pipelines and Automation**

Before looking at available features and features used in related work, we quickly return to the topic of Smart City and how to integrate different data sources and automate the feature engineering process. A popular solution for production-ready ML applications is the use of pipelines that streamline the data pre-processing, feature engineering, potential re-training, and prediction processes. Luckily, all data-sources previously mentioned have APIs or offer download options to get access to recent sensor readings. Next to the  $T_{air}$  readings which can be sourced from Netatmo, Sensor.Community, and/or the DWD, other features are also important to get good prediction, as suggested by Zumwald et al. where  $T_{air}$  at 2m only explained 33% of the prediction [ZKBK21], such as remote sensing information, that for example could be sourced from Google EE.

The actual sensor readings come directly from the sensing layer, in which different external sources could be integrated via API, so that virtual sensors are created that are directly available for example via a overlay network such as SkipNet. In this work, we do not take advantage of other solutions but instead source data directly via API or archives to make a historical data analysis instead of testing real-time applications. After transporting this information via the transportation layer to the data-management layer, the service layer can consume the data streams. An  $T_{air}$  interpolation service could then be placed in the service layer for areal interpolation. This service could then be used for areal interpolation to provide fine-granular  $T_{air}$  maps/grids. The use-case of single station in-

terpolation, for example in combination with moving sensors might be better located in the sensing layer as a virtual sensor; however, challenges would be the dependency on other sensors which might have different sampling intervals.

In this work, we only take advantage of sklearn pipelines<sup>22</sup> in a limited fashion for the actual training and testing process to automatically include scaling and cross-validation and rely on many manual steps to download the data, do pre-processing and QC, and finally extract features and train ML models.

### 4.5.2 Feature Overview

In the following sections, we look at available features for the use-case of  $T_{air}$  interpolation.

#### Essential Climate Variables

Essential Climate Variables (ECV)s are a list of currently 50 variables that are proposed by the WMO to measure climate and climate change. The WMO regularly publishes updates on which climate variables to use and how to measure them [WMO18]. ECVs are generally more focused on measuring climate on a global scale; however, they also contain many variables that are relevant for urban microclimate, such as  $T_{air}$  and land cover. There are three categories of ECVs, namely Atmosphere, Land, and Ocean. And overview of the ECVs can be found online<sup>23</sup>.

Next to official WMO suggestions, other institutions such as the Integrated Climate Data Centre (ICDC) from the University of Hamburg (UHH) also suggest variables to measure climate<sup>24</sup>. Table 4.3 shows the available parameters for which datasets are available.

#### Features used in Related Work

Related studies can give a good overview of which features work best for  $T_{air}$  interpolation. The used features can be roughly divided into 4 categories: raw in-situ measurements, raw satellite data, calculated features, and additional meta data. An important way to use satellite data is to calculate indexes out of the raw data. Alonso and Renard [AR20] used among other data the indexes shown in Table 4.4 to predict  $T_{air}$ . These indexes are either available as precalculated datasets or can be calculated from raw satellite data. Especially the Google EE platform provides a lot of precalculated datasets from various sources such as MODIS [Did21]; however, each index is separately available, therefore requiring some work to combine them into a single dataset. Due to the interference from clouds, these indexes usually also include quality bands, which indicate the quality of the index value for a given pixel, as well as missing values.

<sup>22</sup><https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html>, last accessed: 27.08.2023

<sup>23</sup><https://gcos.wmo.int/en/essential-climate-variables/table>, last accessed: 08.08.2023

<sup>24</sup><https://www.cen.uni-hamburg.de/icdc/data.html>, last accessed: 27.08.2023

Category	Parameters
Atmospheric Data	Air Temperature Pressure Wind Precipitation Clouds Aerosols Humidity Radiation Climate Indices
Ocean	Water Temperature Wave Height (SSH) Salt Content Tide Ocean Colour Climatology Ocean Currents
Ice/Snow	Sea Ice Coverage Sea Ice Thickness Sea Ice Type Snow Thickness (Ice) Snow Water Equivalent (SWE) Land Snow Cover Glacier Thickness Melting Ponds
Land	Albedo Surface Temperature Vegetation Soil Moisture Topography Short-wave Radiation Permafrost
Society	Social Science Parameters

Table 4.3: ICDC Dataset Parameters

In comparison to MODIS, Sentinel satellite data provides a significantly higher resolution at 10-60 m<sup>2</sup> per pixel compared to 500-1000 m<sup>2</sup> per pixel for MODIS but there are no precalculated indexes available for Sentinel data on the Google EE platform.

However, there exist scripts published by other researchers to manually calculate such indexes for example the NDVI index from Sentinel data by the Free University of Berlin<sup>25</sup>.

In comparison to MODIS and Sentinel, many LiDAR datasets are closed source and are not available for research purposes. This is unfortunate as LiDAR data provides a very high resolution of 5-10 cm<sup>2</sup> per pixel and enables the capturing of detailed elevation data. Especially in the context of urban areas and building heights this information can be very useful, for example to calculate the SVF which seems to have a significant impact on T<sub>air</sub> modelling [DRTP19].

Next to index data from remote sensing, there are also other types of information that could be useful for ML applications. Alonso and Renard [AR20] also used the following information:

<sup>25</sup><https://www.geo.fu-berlin.de/en/v/geo-it/gee/2-monitoring-ndvi-nbr/2-2-calculating-indices/ndvi-s2/index.html>, last accessed: 09.08.2023

Variables (Units)	Acquisition Source	Variables (Units)	Acquisition Source
<b>Vegetation Index</b>		<b>Radiation Index</b>	
Normalized Difference Vegetation Index (Difference Vegetation Index (NDVI))	Landsat 8	Spectral Radiance	Landsat 8
Enhanced Vegetation Index (EVI)	Landsat 8	Emissivity	Landsat 8
Soil Adjusted Vegetation Index (SAVI)	Landsat 8	Tasseled Cap Transformation Brightness	Landsat 8
Tasseled Cap Transformation Greenness (GVI)	Landsat 8		
Density of Low Vegetation	LiDAR	<b>Building Index</b>	
Density of Medium Vegetation	LiDAR	Normalized Difference Built-Up Index (NDBI)	Landsat 8
Density of High Vegetation	LiDAR	Urban Index (UI)	Landsat 8
<b>Water Presence Index</b>		Index-based Built-Up Index (IBI)	Landsat 8
Modified Normalized Difference Water Index (MNDWI)	Landsat 8	Building Density	LiDAR
Normalized Difference Water Index (NDWI)	Landsat 8		
<b>Bare Soil Index</b>		<b>Urban Morphology</b>	
Normalized Difference Bareness Index (NDBAI)	Landsat 8	Sky View Factor	LiDAR
Bare Soil Index (BI)	Landsat 8	Standard Deviation (STD) of Building Height	Local Authority
Enhanced Built-Up and Bareness Index (EBBI)	Landsat 8		
Density of Bare Soil	LiDAR	<b>Moisture Index</b>	
		Tasseled Cap Transformation Index	Landsat 8
		Normalized Difference Moisture Index (NDMI)	Landsat 8

Table 4.4: Indexes used by Alonso and Renard [AR20] to predict  $T_{air}$ .

Variable	Data source	Data type	Spatial resolution
Red	Landsat 7,8 and Sentinel 2	Open source	L: 30m, S: 10m
Green			
Blue			
Near infrared			
Short-wave infrared 1			L: 30m, S: 20m
Short-wave infrared 2			
NDVI			L: 30m, S: 10m
IBI			L: 30m, S: 20m
LST	Landsat 7,8		30m
Elevation above sea	STRM		30m
Terrain aspect			
Terrain slope			
Terrain ruggedness			
CHM	LiDAR	Closed source	1m
CHM slope			
CHM aspect			
CHM shadow/SVI			
Building height	LiDAR + building footprint		
Building height sd 1-4m			
Building height sd 4-20m			
Building height sd 20-100m			
Fractional tree cover	LiDAR + ortho-photo		
Tree height			
Distance to coast	Global water occurrence	Open source	30m
Distance to fresh water			

Table 4.5: Features for Air Temperature Interpolation used by Venter et al. [VBEM20]

- Topographic
  - Slope ( $^{\circ}$ )
  - Exposure
  - Curvature
- Land use
  - Distance to railway tracks
  - Distance to points of tourist interest
  - Distance to subway entrances
  - Distances to fountains
  - Water area

For the hyperlocal  $T_{air}$  mapping study done in Oslo by Venter et al. [VBEM20], the used features are shown in Table 4.5.

#### 4.5.3 Features used in this Work

In the following, we describe which features are used in this work for our two use cases, e.g., single station interpolation and areal interpolation. Due to the limited scope of this work, only a subset of the available features will be used to show the feasibility of the respective use-case and get an idea on the upper bounds of RMSE values that can be expected when choosing one of the use cases, as we expect that with an increase of features and therefore availability of information for the ML models to learn from, the prediction quality will increase.

#### Features for Areal Interpolation

$T_{air}$ , relative humidity, atmospheric pressure, precipitation, and wind was sourced from Netatmo and SensorCommunity PWS networks as well as the DWD weather stations as reference data. All remote sensing data acquired in this work was processed using the Google EE [GHD<sup>+</sup>17] as it offers a unified way of accessing data and offers enhanced processing capabilities that are especially important when dealing with these large datasets that can grow as large as several hundred terabytes. The following datasets have been used and downloaded from the Google EE platform:

- MODIS/061/MOD13A1: MODIS Vegetation Indexes NDVI and EVI (500m, 16 days) [Did21]
- COPERNICUS/DEM/GLO30: Copernicus Digital Elevations Model (30m) [cop]

Potential datasets that could be used in the future are:

---

Measurement (Units)	Spatial Resolution	Temporal Resolution	
<b>Weather Station/Sensor Measurements</b>			
Air temperature (°C) Mean	Single location	10 min (Sensor.Community) 10 min (DWD) 30 min (Netatmo Historical) 10 min (Netatmo Live) 10 min (Sensor.Community) 10 min (DWD) 30 min (Netatmo Historical) 10 min (Netatmo Live) 10 min (Sensor.Community) 10 min (DWD) 30 min (Netatmo Historical) 10 min (Netatmo Live) 10 min (Sensor.Community) 10 min (DWD) 30 min (Netatmo Historical) 10 min (Netatmo Live) 10 min (Sensor.Community) 10 min (DWD) 30 min (Netatmo Historical) 10 min (Netatmo Live) 10 min (Sensor.Community)	
Relative Humidity (%)	Single location	10 min (DWD) 30 min (Netatmo Historical) 10 min (Netatmo Live) 10 min (Sensor.Community)	
Atmospheric Pressure (mBar)	Single location	10 min (DWD) 30 min (Netatmo Historical) 10 min (Netatmo Live) 10 min (Sensor.Community)	
Wind Strength (km/h)	Single location	10 min (DWD) 30 min (Netatmo Historical) 10 min (Netatmo Live) 10 min (Sensor.Community)	
Wind Direction (°)	Single location	10 min (DWD) 30 min (Netatmo Historical) 10 min (Netatmo Live) 10 min (DWD)	
Precipitation (mm)	Single location	30 min (Netatmo Historical) 10 min (Netatmo Live)	
<b>Remote Sensing Data</b>			
NDVI (MODIS)	500m	16 days	
EVI (MODIS)	500m	16 days	
DEM (Copernicus)	30m	2015 - 2017	

Table 4.6: Features for Air Temperature Interpolation Used in this Work

- MODIS\_061\_MOD15A2H: MODIS Leaf Area Index/FPAR  
(500m, 8 days) [MKP21]
- COPERNICUS\_S2\_SR: Sentinel-2 Multi Spectral Instrument, Level-2A  
(10m, 5 days) [sen] (for manual index calculation)

Additionally, location data was incorporated into the models by longitude and latitude values in coordinate reference system EPSG:4326. As noted by Hengl et al. this might not be optimal [HNW<sup>+</sup>18] and could be improved by using distances between sampling locations instead. The features used in this work are shown in Table 4.6 and were chosen based on availability and relevance for  $T_{air}$  modelling. The number of features in the initial scope of this work is quite limited and could be increased to gain better prediction results.

### Features for Single Station Interpolation

For the single station interpolation use-case, we keep the setup rather simple. The idea behind this approach is the hypothesis that the model does not really care about the location of the sensor, but only about the similarity in temperature curves. Therefore, we use the  $T_{air}$  readings of neighbouring sensors as covariants ordered by the distance to

the sensor and always in the same order. Next to the  $T_{air}$ , we also include the humidity and pressure of neighbours as covariants in the same order, and the time of the reading converted to a sinus function, so that we can explore if the time of day has an influence on prediction errors as shown in Appendix 3.

## 4.6 Additional Considerations

Next to the features, there are also additional considerations when creating train and test datasets. These include sampling, normalization, and scaling, dealing with correlations, and more. If your model for example expects uncorrelated input variables, one could calculate the variance inflation factor (VIF) to make sure the input variables are uncorrelated. If the VIF is over 10 [MPV21] or more restrictive over 3 [ZIE10], this indicates that the model is invalid. Another option could be principal component analysis to turn correlated variables into new uncorrelated ones. In this work, we try to choose models which can deal with correlations naturally like RF, so we reduce the complexity of working with this type of features.

Other considerations include imputation for missing values and feature scaling. These will be covered in the evaluation where appropriate, as some models such as KNN need feature scaling, while others such as RF or HistGB do not need scaling, as all models except HistGB cannot handle missing values and need imputation.

### Spatial Autocorrelation

Spatial autocorrelation is the presence of ‘systematic spatial variation in a mapped variable’ [Hai01]. If adjacent variables tend to have similar values, spatial autocorrelation is positive. In contrast if adjacent variables tend to have very contrasting values, the spatial autocorrelation is negative. It can be defined traditionally via the Moran’s I index [Mor48] or the Geary’s coefficient [Gee54]. The goal of using ML in this context would be that these correlations can be ignored, simplifying the interpolation process. Especially in geostatistical methods such as Kriging, not dealing with correlations can lead to wrong interpolations and the introduction of bias.

### Temporal Autocorrelation

Next to spatial autocorrelations, there can also be temporal autocorrelations between successive values of the same variable. Also, there can be seasonal trends that might need to be corrected, as well as temporal lag. In the case of temperature sensors, this could relate to the time it takes for a sensor to adjust to a new temperature. A reference-grade sensor might adjust in a few seconds, where a LCS might need more time. Especially with Netatmo sensor which have a suboptimal ventilation and small form factor, this could

mean that after they heated up, they need more time to cool down again, resulting in a bias towards longer hotter temperatures.



## 5 Evaluation

The goal of this chapter is to evaluate different use cases for ML-based interpolation in the context of  $T_{air}$  interpolation. In Chapter 3, different models for ML-based interpolation were already introduced. Afterwards, the available data and features were introduced in Chapter 4. In this chapter, the different models are compared with different datasets and features to test the feasibility of two main use cases:

1. Interpolation of  $T_{air}$  for a specific location
2. Areal interpolation of  $T_{air}$

Next to these main uses cases we discuss there are other ways the different regression models can be used, primarily decided by the features used. Some regression models could for example be used to predict future  $T_{air}$ , i.e., extrapolation, as well; however, this is out of the scope of this work.

### Implementation Details

All ML models are implemented using Python and the scikit-learn library [PVG<sup>+</sup>11]. Additionally, the following libraries are used:

- Pandas, Numpy, Geopandas, scipy, matplotlib, shapely, contextily, pytz, sklearn, seaborn, rasterio, polars, Google Earth Engine, pykrige, pytorch, missingno

### Validation Methodology

To validate the different models, we use the following methodology:

We evaluate two locations based on data availability, e.g., Hamburg and Stuttgart, that also have slightly different climate characteristics, although both are located in Germany in a moderate cool climatic zone, with Hamburg located near the coast in rather Maritim climate and Stuttgart located more inland in a continental climate. Hamburg has also higher precipitation and wind compared to Stuttgart. For both locations, we collected PWS data from Netatmo and Sensor.Community; however, Netatmo data is only available for the whole month of June 2023 for Hamburg and for a single timestep on 19. June 2023 14h for Stuttgart, while Sensor.Community data is available for the whole months of January and June 2023 for both locations.

For both locations for both interpolation use cases, we first compare all available models, e.g., Linear Regression, KNN, RF, SVR, and HistGB, with all features against each

---

other to get an understanding of the overall performance of the models and maximum achievable potential, given the assumption that more features generally improve prediction quality. The datasets are split into a training and test set. 70% of the data is used for training and 30% is completely withhold for testing. The training and test set are split randomly. The same split is used for all models and sensors to ensure comparability. Afterwards, the most promising model with the lowest error is used to further evaluate specific influences such as distances between neighbours or the amount of training data and time intervals. For the further evaluation, 5-Fold Cross Validation is used to reduce the risk of overfitting [K+95]. We note here that a k of 10 would be preferred; however, due to the computational overhead we only use a k of 5 for exploration.

All ML models in the initial evaluation are trained using the default parameters as defined in the scikit-learn library. The only exception is the number of estimators for the RF Regressor and the number of iterations for the HistGB Regressor, which are both increased from 100 to 200 to improve the performance of the models as found out by previous exploration. To get the best possible performance of each model, an exhaustive grid search would be preferred for all models to fine-tune all hyperparameters; however, due to the computational overhead of running exhaustive grid searches, this is not feasible in the scope of this work.

For the error metrics, the following candidates are available: MSE, MAE, RMSE,  $R^2$ . MSE is the most used error metric and is more sensitive to outliers, whereas MAE punishes outliers less but is therefore also more variable. Related work commonly uses RMSE as it has the same unit as the response variable, therefore RMSE is used as the main error metric.  $R^2$  is additionally used to get an understanding of the variance of the model.

## 5.1 Interpolation of Air Temperature for a Specific Location

The first use-case to be evaluated is the interpolation of  $T_{air}$  for a specific sensor. The main idea behind this approach is to use ML models to capture the dependency between neighbouring weather stations and sensors, so that in case a sensor is not available, the  $T_{air}$  can be interpolated more easily, especially over a longer period. Another use-case could be that a sensor location is not stationary, and the sensor for example moves through a city mounted on a bus or bike. In this case, the sensor could be used to capture a snapshot of the  $T_{air}$  at a previously unobserved location, increasing the spatial coverage of the sensor network. The evaluation for a specific location is done using the following steps:

1. Create datasets for training and testing based on the datasets for June 2023 after QC for Hamburg and Stuttgart
  2. Data pre-processing for the specific models. Only HistGB supports missing values, therefore we need to fill in missing values for all other models using an imputer. We use a simple mean imputer for this purpose, that fills missing values with the mean
-

value for each row, i.e., each timestamp, not column. All input data is normalized using a standard scaler as some models such as KNN need normalized data, which should not have any impact on models that don't need normalization of input features such as RF and Gradient Boosting. This pre-processing step could be further improved by using different means of imputation or using different scalers.

3. Fit each model using the datasets and evaluate the performance using all available features for all locations to get an overview of the performance of each model. The dataset is split 70% for training and 30% for testing, but no k-Fold Cross Validation is used due to the computational overhead for the initial comparison across all sensor locations.
4. Choose the most promising model and further evaluate the influence of different features, e.g., distances between neighbours, the amount of training data, and time intervals. This step only selects a subset of stations of particular interest, e.g., in the city centre or near water and adds 5-Fold Cross Validation for further trust in the results.

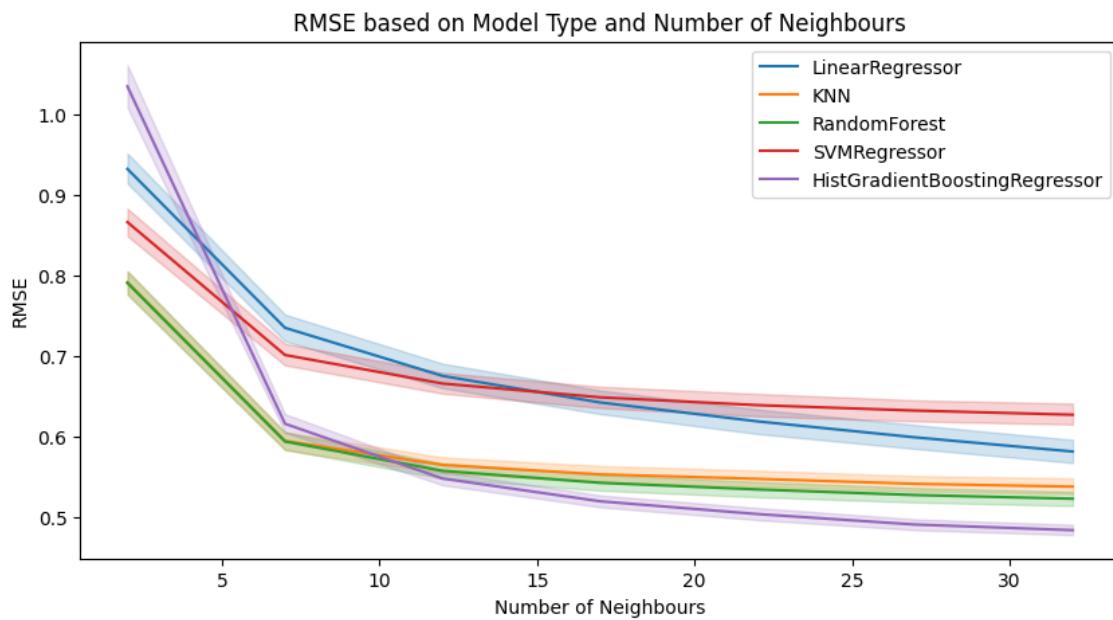


Figure 5.1: RMSE by Model Type with the Confidence Interval of 95%

### 5.1.1 Model Comparison

First, we compare the overall performance of the models against each other. Figure 5.1 shows the RMSE of each model across both Hamburg and Stuttgart for the month of June 2023, while a more detailed overview can be found in Appendix Figure 2 and 3. For all models, one evaluation was done using the imputed dataset without missing values, as

Linear Regression, KNN, RF and SVM Regression cannot handle missing values. Gradient Boosting was once run with and once run without imputed data, where the imputed data performed better at 2 neighbours, possibly due to neighbours with missing values, but performed slightly worse than not imputed data for higher amounts of neighbours. SVM, KNN and RF seem to hit a plateau when reaching a certain number of neighbours, around 15-20, whereas Linear Regression and Gradient Boosting continue to improve with a higher number of neighbours.

Gradient Boosting shows the lowest RMSE starting with over 10 neighbours and has the smallest confidence interval. The non-imputed data also performs slightly better, as seen in Appendix Figure 3, therefore we will continue a more detailed evaluation with the Gradient Boosting model without imputed data. It is to note, that the individual models were mainly run with default parameters as given by sklearn library, therefore the performance of the individual models could be further improved. The exact parameter configurations can be found in Appendix 2.

### 5.1.2 Further Evaluation - Gradient Boosting

HistGradientBoostingRegressor without imputation achieves overall the lowest RMSE starting with more than 10 neighbours and continuously improves with more neighbours; however, the improvements after 30 neighbours are small. This model is great because it has native NaN support and we can save the imputation step that can possibly introduce bias into the model. The difference to KNN and RF is not that big with a little less than RMSE of 0.1; however, the confidence interval is also smaller compared to the other models. Due to the lowest error and performance benefits compared to other models such as RFs and no need to use imputation, the HistGradientBoostingRegressor is used in the following section to further investigate feature importance, the influence of different time intervals and the impact of QC on the process.

#### Influence of Distance between Neighbours

The goal with hyperlocal temperature mapping is to get a deep insight into very local climatic conditions. If two sensors are located in the same climatic conditions, in an extreme case directly situated next to each other, both sensors should measure the same temperature. Therefore, the hypothesis is, that closer stations have a higher influence on the prediction quality, e.g., if closer stations are chosen as neighbours the prediction quality is better compared than if the same number of neighbours is chosen but further away. We investigate this hypotheses by comparing the RMSE of prediction quality for different distances between neighbours. Because there are not so many stations available for Hamburg, that for example 10 stations could be selected in a 500m radius, we look at the minimum distance instead and remove neighbours that are too close to the station.

---

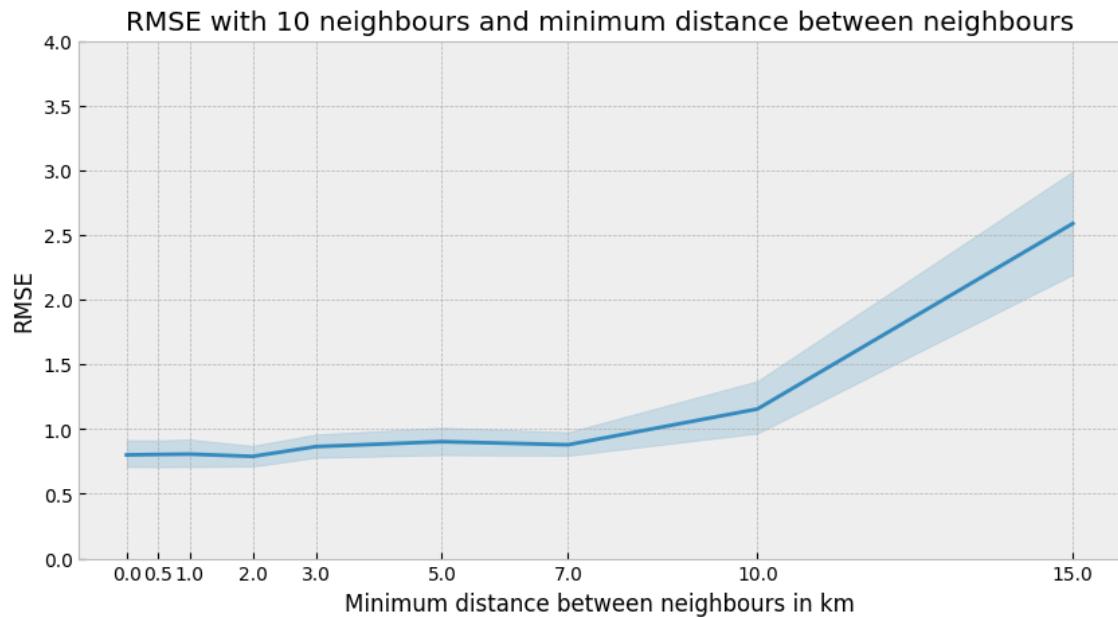


Figure 5.2: RMSE for Increasing Minimum Distance with 10 Neighbours, Hamburg, Netatmo

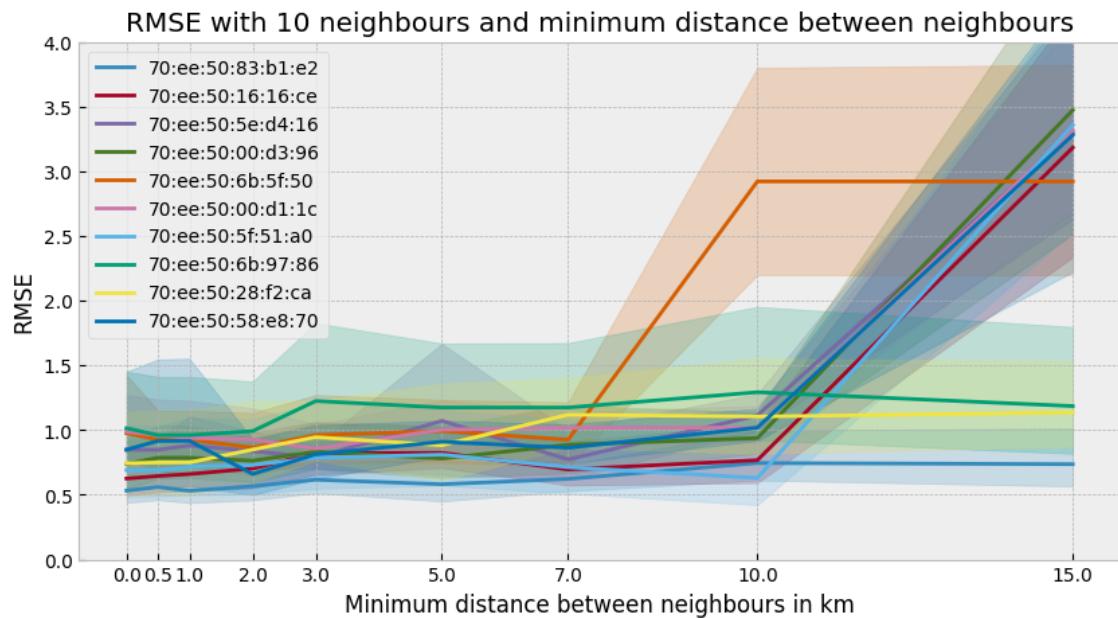


Figure 5.3: RMSE for Increasing Minimum Distance with 10 Neighbours By Station Id, Hamburg, Netatmo

The evaluation is done using 5-fold cross validation for a subset of 10 stations in Hamburg, spread across a bigger area so we can cover different distances between neighbours and different climatic conditions, e.g., distance to water, situated in the city centre etc. The list of stations can be found in Appendix 4. The number of neighbours was set to 10 as this number was the number at which Gradient Boosting outperformed other models. Figure 5.2 shows the RMSE for different minimum distances between stations in km for 10 stations from Netatmo in Hamburg with 10 neighbours. We can see that the RMSE is increasing minimally across the first few kilometres and then starts to sharply increase with greater distances over 10km. However, if we take a look at Figure 5.3 which shows the same data but grouped by station id, we can see that there are several stations whose RMSE does not increase with greater distances.

## Feature Importance

Next to the temperature, there are also other features that could be included in the prediction process such as the time, humidity, pressure, etc. In order to understand how much the features contribute to the overall prediction quality, we first compare the RMSE of the model with only the temperature as input feature for each neighbour, and then test with adding time, humidity, pressure, and finally all features combined. Time is only a single column of the transformed timestamp, whereas humidity and pressure are added for each neighbour. In the end the input looks as follows:  $[ta_1 \dots ta_n, \text{time}, \text{humidity}_1 \dots \text{humidity}_n, \text{pressure}_1 \dots \text{pressure}_n]$

Figure 5.4 shows the RMSE for different features for 10 stations in Hamburg with 30 neighbours. We can clearly see, that there is no visible difference between only temperature and all features combined, therefore in Figure 5.5 we take a look at the permutation importance of a single station for temperature and time on a 30% test set. The values are calculated based on the trained regressors from the 5 different folds of the cross validation; however, the test set was always the same and not the same used during the cross-validation, due to technical limitations in the sklearn library. However, the results indicates that only a very small number of stations, in this case the neighbours 4 and 13, have a high influence on the prediction quality. All other stations have a very low influence on the prediction quality, with time having no visible influence at all. This is a very interesting result, as it indicates that the model only needs a handful of neighbours that have similar temperature distributions to the target station in order to make a good prediction. Other features such as time, humidity and pressure do not seem to have any influence on the prediction quality, and could therefore be ignored, saving computational resources and making the model more robust. This assumption could be further validated by looking at more stations and their permutation importance.

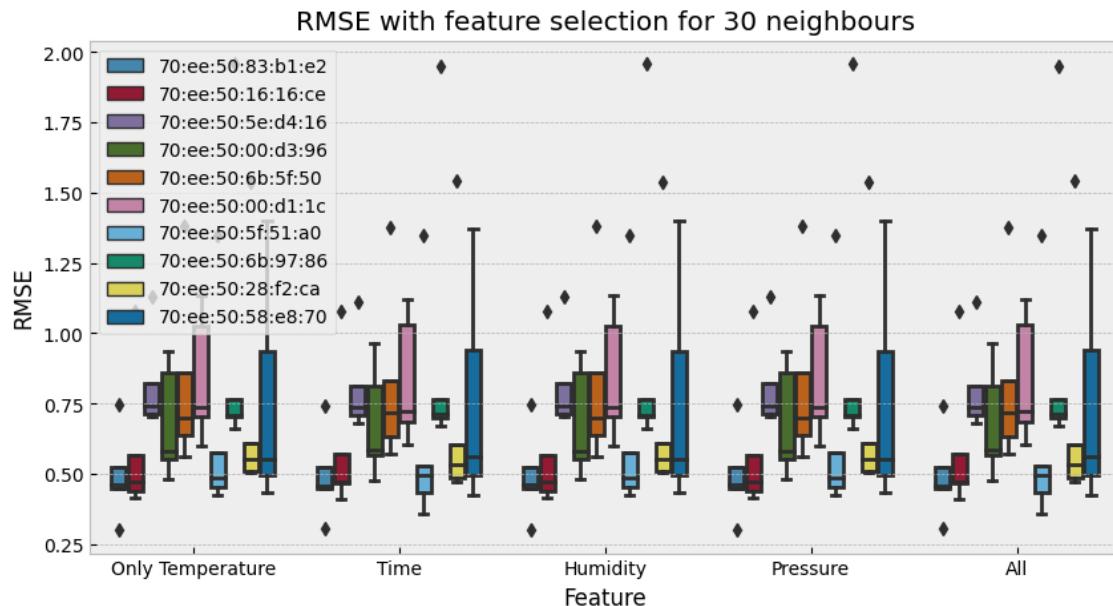


Figure 5.4: RMSE based on Features Selected with 30 Neighbours, Hamburg, Netatmo

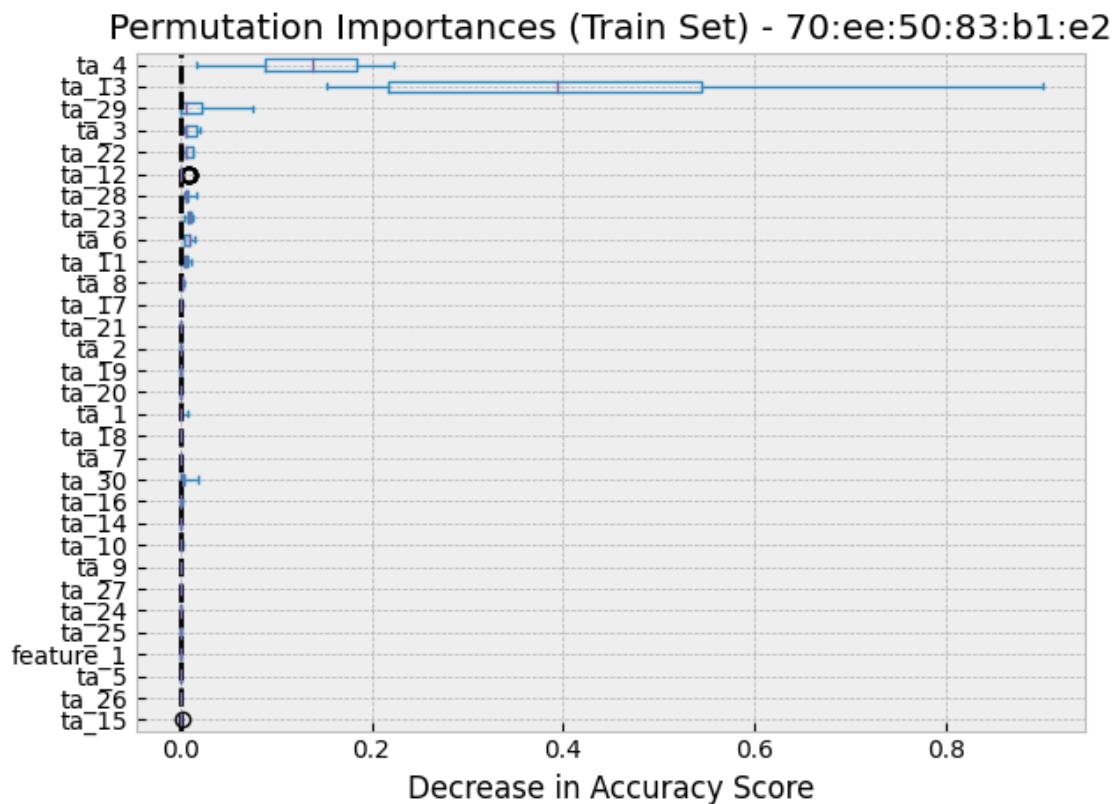


Figure 5.5: Permutation Importance for Single Station on 5-Fold Cross Validation, Hamburg, Netatmo

### Influence of Quality Control

In this work, we decided to use QC steps to exclude stations that are either faulty, setup indoors, or perhaps setup in an incorrect way (i.e., too close to a wall or the ground). However, depending on the use-case, one might want to include sensor readings that are flagged as outliers by the m5 buddy check either because the sensor density is lower or one would actually want to get a more fine-granular overview of hotspots. As a result, we compare the different QC levels of m4 and m5, i.e., Temporal Correlation and Spatial Buddy Check, to see if the QC levels have an influence on prediction quality of the interpolation process.

For this test, we use HistGB without imputed data with 30 neighbours across our 10 testing stations from Netatmo from June 2023 in Hamburg together with 5-fold Cross Validation. The result shown in Figure 5.6 show the smallest RMSE for the m4 QC level, i.e., the Temporal Correlation, which could be due to the increase in data availability per station to train on. This result indicates, that maybe the m5 buddy check is too strict and maybe the QC level m4 could be sufficient for the interpolation process.

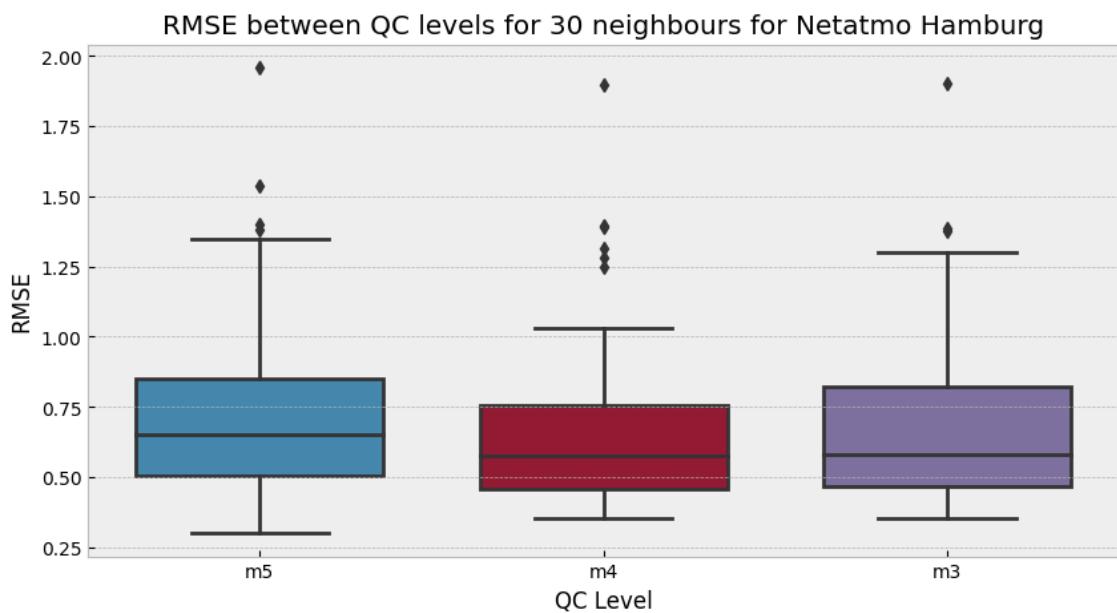


Figure 5.6: QC Level Comparison for Netatmo Data for Hamburg, June 2023

### Influence of Time Intervals and Non-Stationary Sensors

Sensor networks have the advantage, compared to remote sensing approaches, that they provide high temporal resolutions; however, at the cost of spatial coverage. In the context of  $T_{air}$  sensing,  $T_{air}$  sensors could be mounted to busses, cars, scooters, or bikes to increase the spatial coverage of the sensor network. Unfortunately, there are currently no datasets of actual moving sensors available; however, we can simulate a non-stationary sensor by removing data points of specific sensors in either a fixed interval, e.g., simulating a bus

line that visits a certain location in a rather fixed interval, or randomly, simulating a bike or scooter that is used periodically.

We have already indirectly simulated missing data by applying QC steps before using the station data to train the models, as seen in Section 4.4, where we lose quite some data. An interesting exploration could be to use the previous sensor with Netatmo Station Id `70ee:50:83:b1:e2` and choose neighbours that have a high influence on the prediction quality, e.g., neighbour 4 and 13, and remove data points in both fixed and random intervals to simulate a moving sensor instead of a stationary sensor. We could also check what happens, if not only two but multiple neighbours are turned into simulated moving sensors.

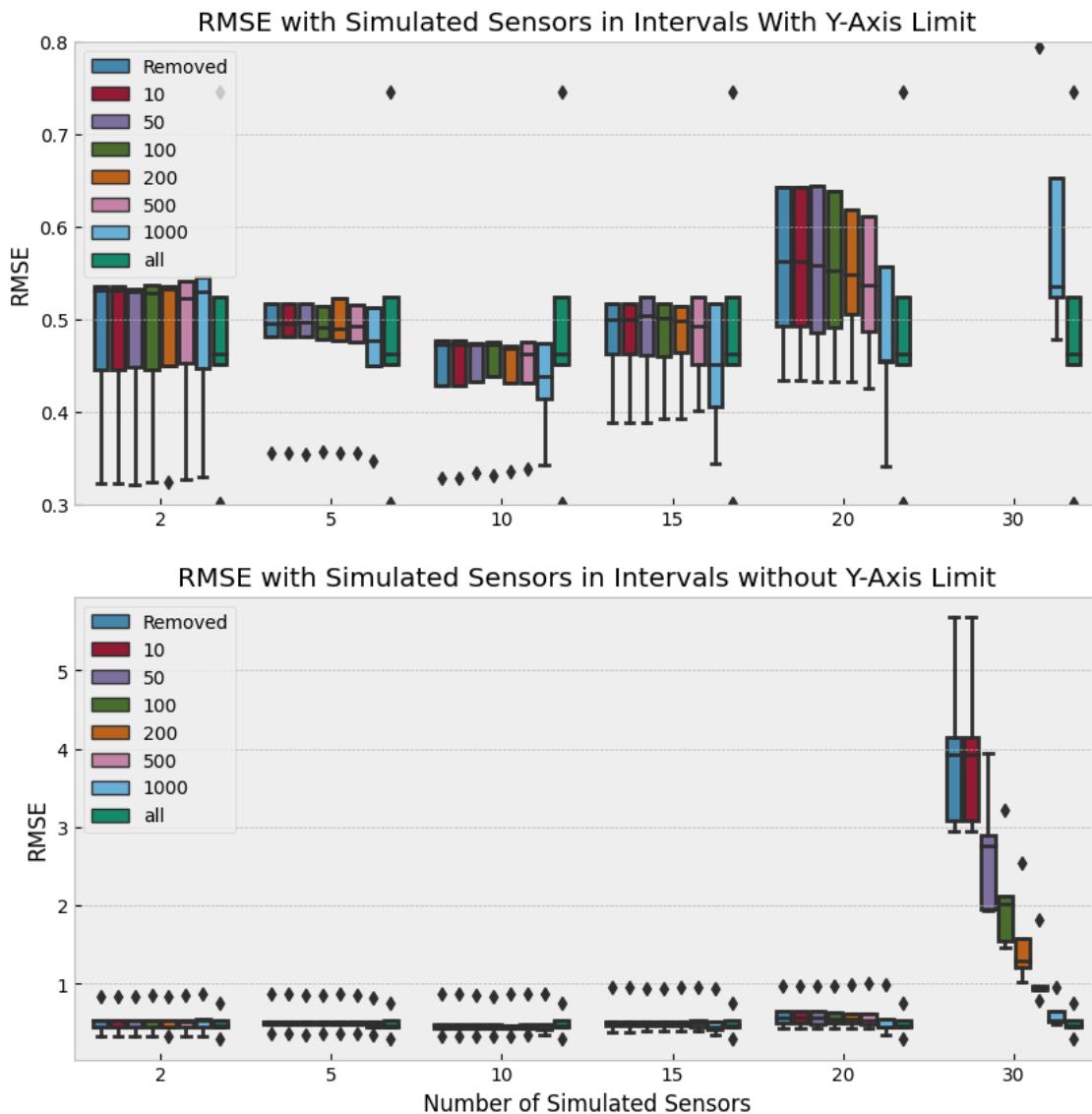


Figure 5.7: Comparison between amount of data sampled from simulated moving sensors for Netatmo station `70ee:50:83:b1:e2` with 30 neighbours

In Figure 5.7, results are shown for the Netatmo station `70ee:50:83:b1:e2` with 30 neighbours where we start off with simulating 2 sensors, i.e., `70:ee:50:3f:1f:1c` and `70:ee:50:96:d9:70`, which had high permutation importance, to quickly see differences between different sampling size for individual sensors. Iteratively, additional sensors were added from a list of randomly sorted neighbouring station ids, so that in each iteration the stations from the previous iteration are included. The ‘Removed’ columns show the results with the moving sensors completely removed, while the ‘all’ column shows what happens if all available sensor readings are used, as reference. The other numbers show the number of readings that are randomly sampled for that sensor. The number of available sensor readings per sensor is averaging around 1000-1500 readings per sensor, which was reduced from the raw data due to QC or missing data. 5-fold cross validation was used for each combination of number of moving sensors and number of readings to sample. We can see, that a low number of moving sensors, especially with high number of readings available, do not have a high influence on the RMSE, which averages between 0.5 and 0.4; however, starting with 20 moving sensors, the RMSE jumps a lot, especially with 30 moving sensors, i.e., all sensors are moving and have as little as 10 readings for the whole month of June in Hamburg. This exploration suggests, that there is not a big difference between stationary sensors and moving sensors with less frequency, as long as a certain amount of readings for that location is available and not all sensors are not available. Further research in this direction could focus on the comparison between different stations with varying number of neighbours and sample sizes. Additionally, we didn’t test with fixed intervals yet, so this could also be an interesting exploration to for example simulate bus lines with sensors.

## 5.2 Areal Interpolation of Air Temperature

The second use-case for ML-based interpolation of  $T_{air}$  is the areal interpolation, e.g., turning a set of single data points into a continuous temperature layer/grid. The problem with this approach is that in comparison to interpolating a single sensor, there are potentially many locations that have no sensor data and therefore no target variable to be trained with. The main assumption here is that locations with similar features, e.g., land coverage, soil temperature, SVF, solar radiance, etc. have similar  $T_{air}$ . This assumption is one of the motivating factors behind LCZ research. The main challenge here is to find the right features that capture the dependency between the different locations and choose how to model spatial information.

In the following section, areal interpolation of  $T_{air}$  is explored by comparing the RMSE between different models, previously discussed in Chapter 3. We primarily focus on the area of Stuttgart, as SensorCommunity data for this area was much easier to acquire, as seen in Figure 5.9. We focus again on the month of June 2023, as it contained a lot of periods of prolonged hot weather, as seen in the Appendix Figure 5, and we are in-

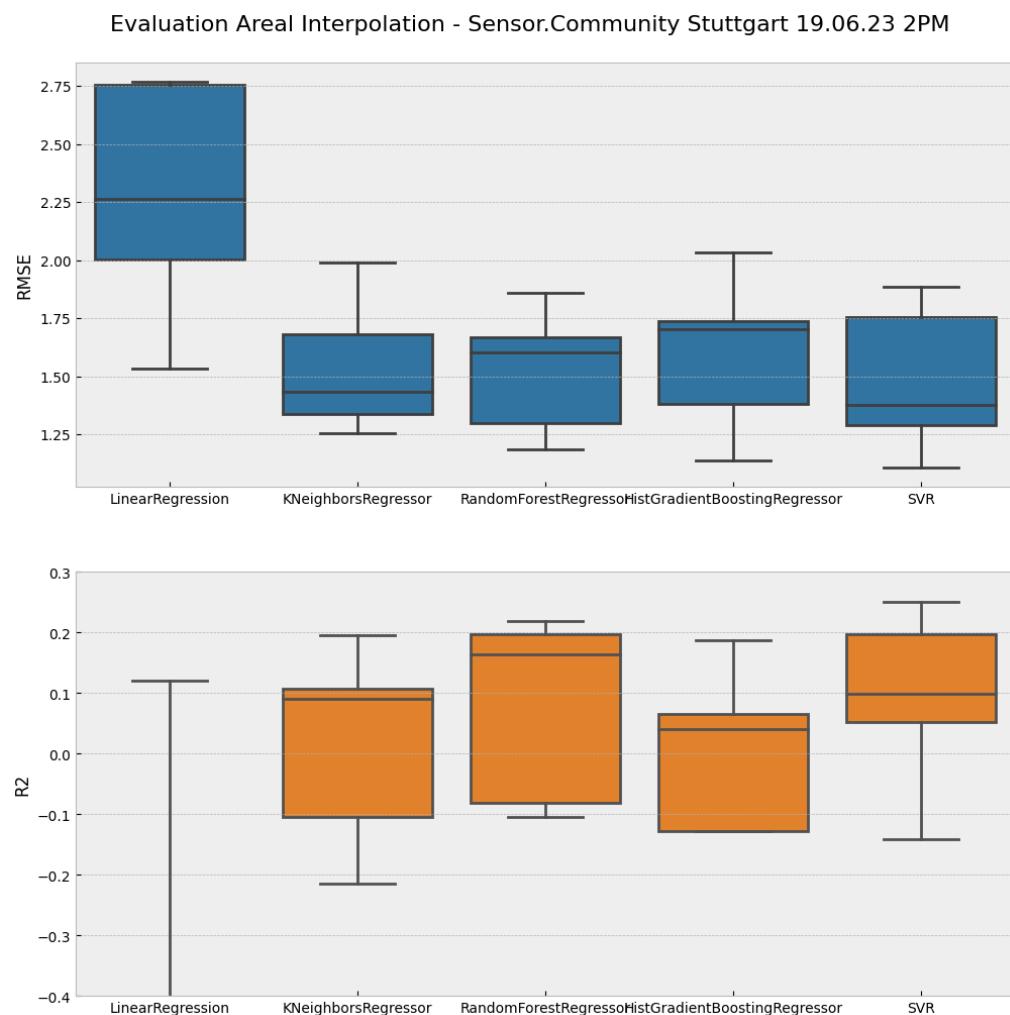
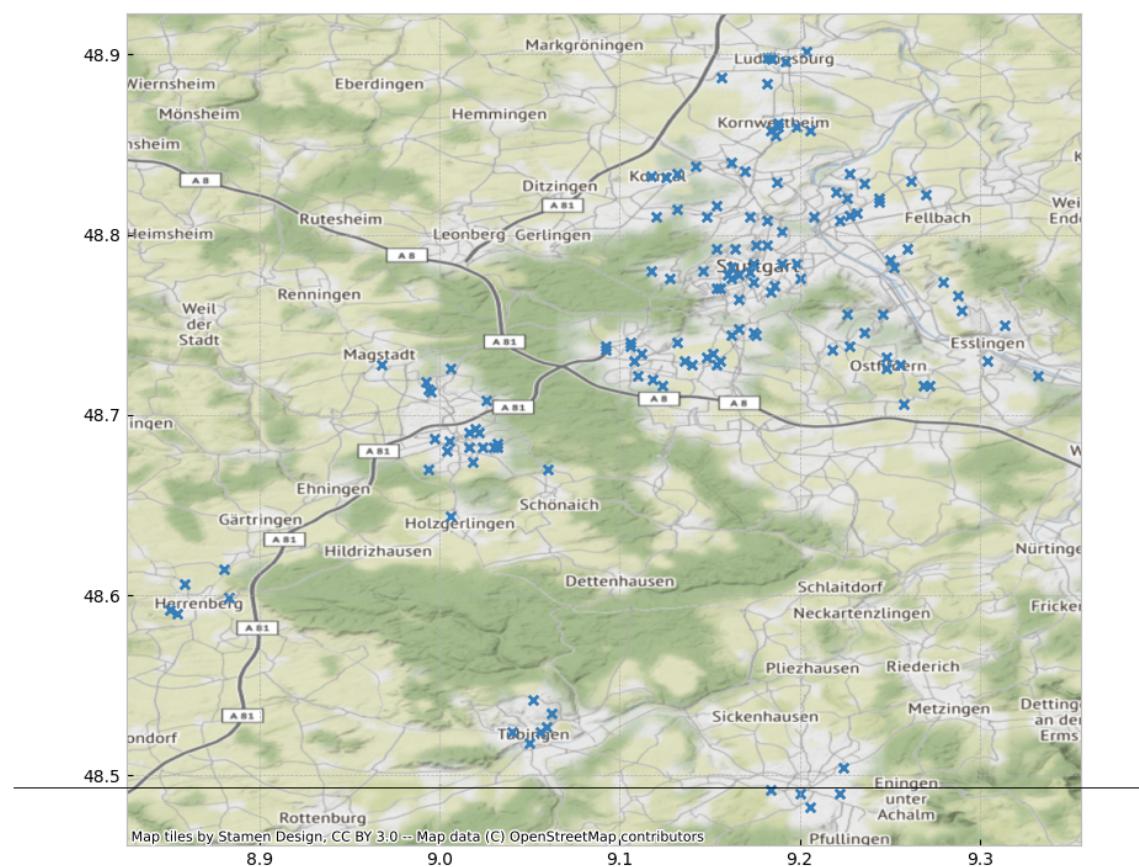


Figure 5.8: Areal Interpolation Comparison, Sensor.Community Stuttgart, 19.06.2023 2PM



terested in mapping  $T_{air}$  especially during hot periods. Similar to Kriging, in our areal interpolation exploration, we focus on the interpolation of  $T_{air}$  for a single timestamp. Our approach could be extended in the future, to also include a temporal aspect as well; however, this is out of the scope of this work. Based on the DWD reference station, we chose the 19.06.2023 as our reference data to try out several types of areal interpolation. To get a good understanding, what kind of  $T_{air}$  to expect that day, we also took a look at the 2m and 5cm  $T_{air}$  at the DWD station on the 19.06.2023, as seen in Appendix Figure 6. Interest to note here, is that the 5cm  $T_{air}$  fluctuates way more than the  $T_{air}$  at 2m, indicating that height information of a sensor place an important role, especially in areal interpolation where you cannot only rely on the relationship between neighbouring temperature curves but need to capture placement information as well.

### 5.2.1 Model Comparison

To compare the different models, we first need to define at which time we want to test our model, as we can only interpolate a single timestamp at a time. We chose the hottest temperature for the day at 2PM as well as the coldest point that day at 5AM. We start of with only Sensor.Community data and the following features:

- Longitude, Latitude
- Humidity (where available)
- Pressure (where available)
- Indexes (500m) -> NDVI, EVI
- DEM
- Temperature and Distances to the closest 30 neighbours (similar to RFSP)

Each model, i.e., Linear Regression, KNN, RF, HistGB, SVM, was run with default sklearn parameters. We used 10-fold cross validation with a StandardScaler to evaluate the model performances. The results can be found in Figure 5.8. We can see a RMSE between 1.7 and 1.3 for KNN, RF, HistGB, SVM, while Linear Regression clearly falls behind with over 2. The  $R^2$  score of all models is rather low with 0.2 maximum and in some cases even negative values, showing high variability during especially hot weather.

### 5.2.2 Comparison with Kriging

In order to evaluate the performance of the ML model, we need to first get a better understanding of the interpolation quality of existing interpolation techniques. Next to simpler deterministic interpolation methods, such as IDW that are simple but performant, yet

---

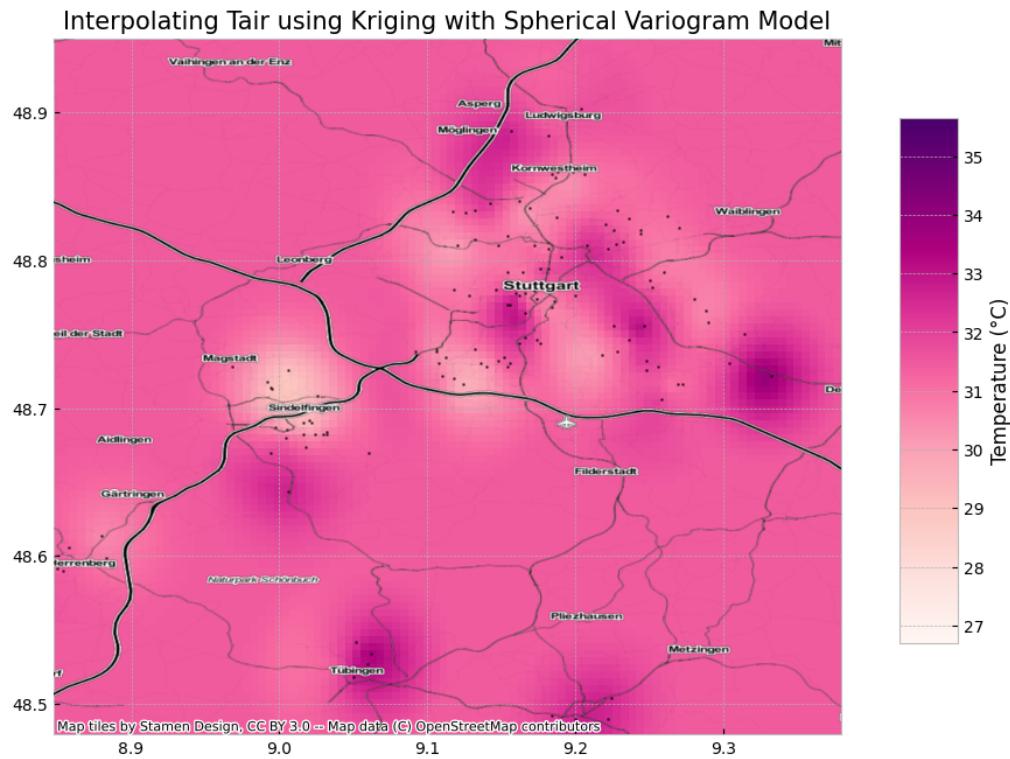


Figure 5.10: Ordinary Kriging Spherical Variogram, Sensor.Community Stuttgart, 19.06.2023 2PM

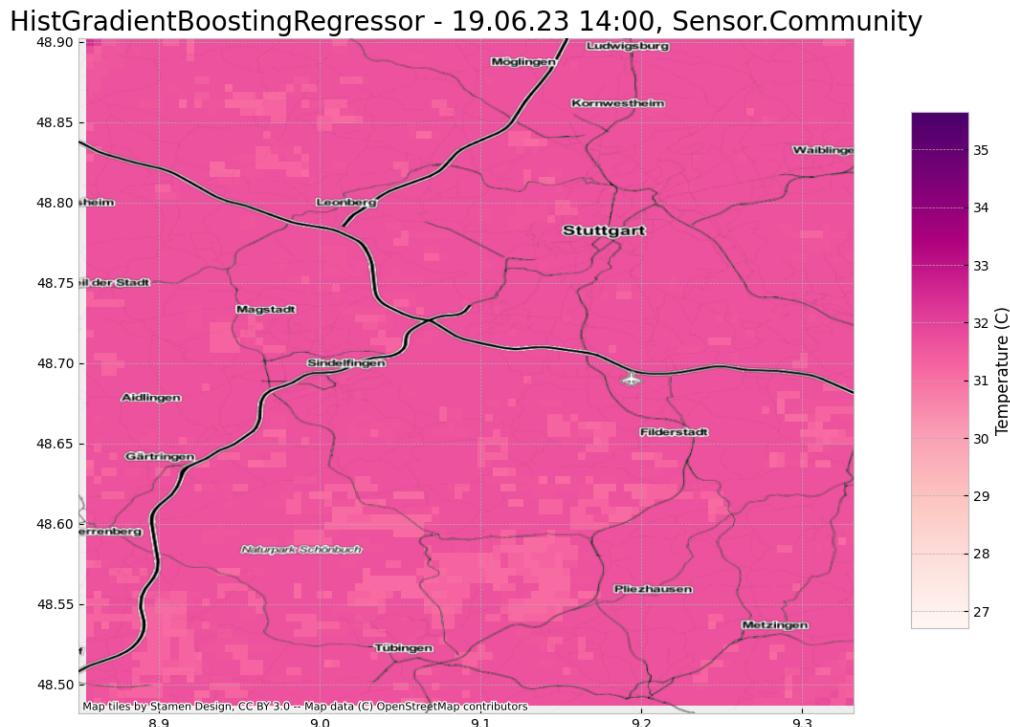


Figure 5.11: HistGradientBoostingRegressor Areal Interpolation, Sensor.Community Stuttgart, 19.06.2023 2PM

struggle to capture more complex interdependencies, there are also more complex methods available. The most common geostatistical method for interpolation is Kriging. In the scope of this work, we unfortunately cannot compare available Kriging methods with each other and therefore need to focus on a subset of methods, namely those easily available via the pykrige Python library, i.e., Ordinary and Universal Kriging. We note however, that for example according to [NAEB23], there are many other more sophisticated and specialised Kriging methods such as EBK and EBKRP that could perform better. For OK, we use the same dataset we used previously but only use the coordinates and the target  $T_{air}$ . With a 80% train and 20% test split, we achieved the following RMSE and  $R^2$  scores in Table 5.2.2. If we choose the same data and train and test our ML models on it, we get the RMSE and  $R^2$  shown in Table 5.1.

Variogram Model	RMSE	$R^2$
linear	1.712089	-0.001049
power	1.712089	-0.001049
gaussian	1.327896	0.397814
spherical	1.272819	0.446732
exponential	1.471249	0.260778

Table 5.1: RMSE and  $R^2$  of Ordinary Kriging for Stuttgart 19.06.2023 2PM, Sensor.Community

ML Model	RMSE	$R^2$
KNeighborsRegressor	1.314883	0.183336
RandomForestRegressor	1.205100	0.314014
HistGradientBoostingRegressor	1.098671	0.429831
SVR	1.432225	0.031072

Table 5.2: RMSE and  $R^2$  of ML Models for Stuttgart 19.06.2023 2PM, Sensor.Community

From these results we can see, that our ML models outperform OK in terms of RMSE but lack behind in  $R^2$  score. To show the differences, we use both Kriging and HistGB to interpolate a map. The results of the best performing model according to  $R^2$  score, i.e., the spherical variogram, can be seen in Figure 5.10. The HistGB results can be seen in Figure 5.11.

### 5.2.3 Comparison during Night

Next to mapping  $T_{air}$  when it's especially hot, we also take a look at night time  $T_{air}$  mapping to have a baseline for comparison and try to understand if there are differences. The results from the 10-fold cross validation on the same dataset but with the time set to 5am, i.e., the time of that day the  $T_{air}$  was lowest, can be found in Figure 5.12. The HistGB model was then trained again with the error values in Table 5.2.3 and the resulting map

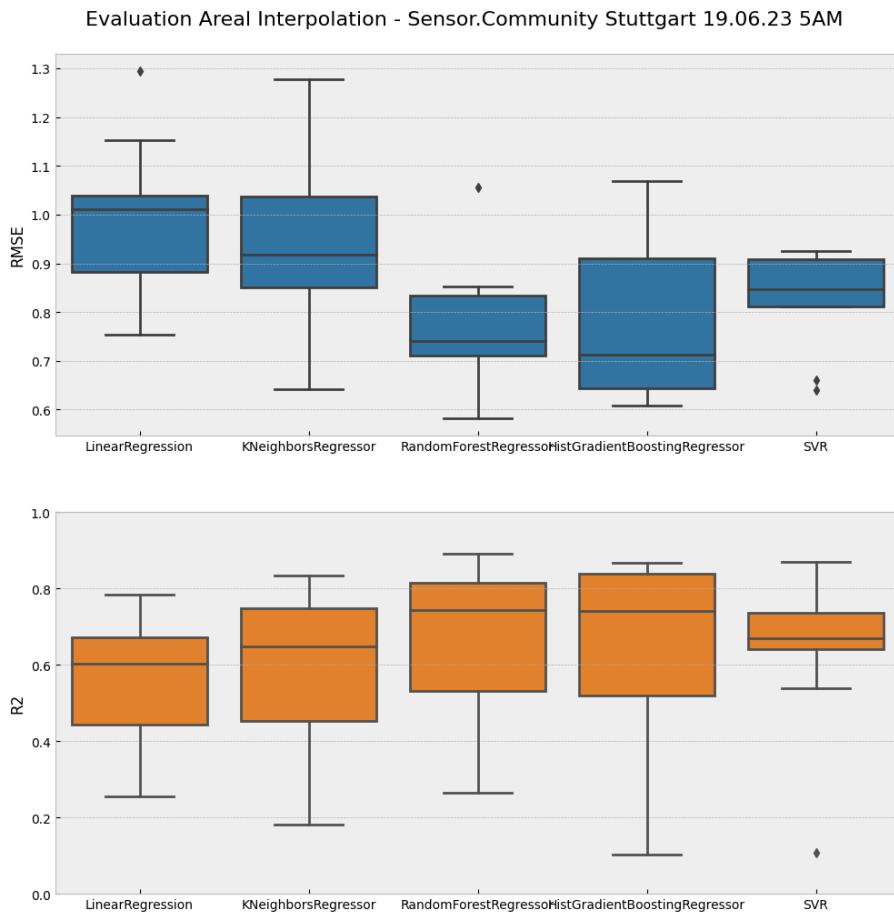


Figure 5.12: Areal Interpolation Comparison, Sensor.Community Stuttgart, 19.06.2023 5AM

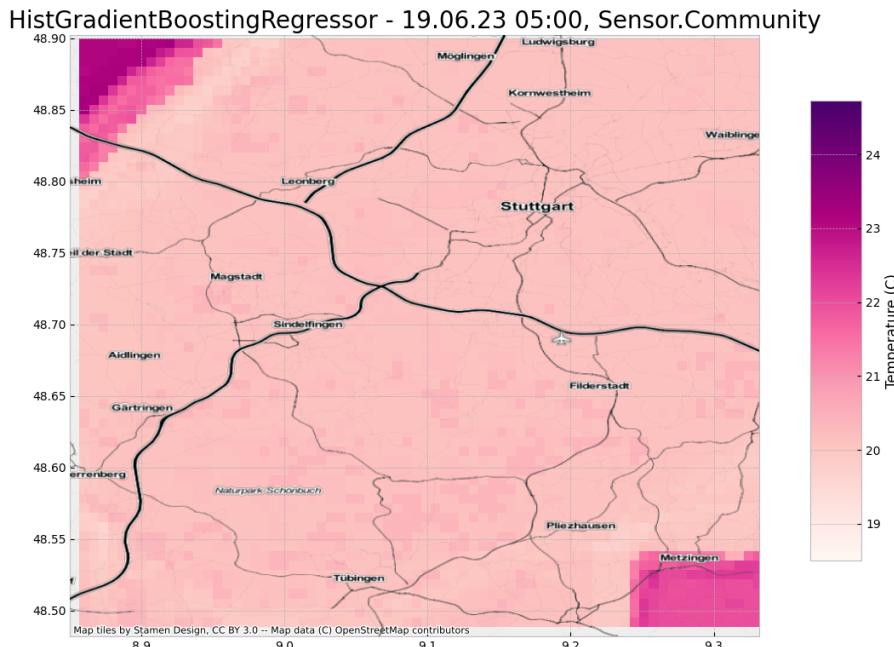


Figure 5.13: HistGradientBoostingRegressor Areal Interpolation, Sensor.Community Stuttgart, 19.06.2023 5AM

in Figure 5.13.

We can clearly see, that  $R^2$  values during the night are way higher, suggesting that we miss some kind of information during the day, such as solar radiation and LST, which might need to be integrated into our model for better prediction results. Additionally, we could also improve the accuracy by using Sentinel 2 and Landsat readings due to their way higher spatial resolution, as currently the MODIS remote sensing values have a 500m pixel size in comparison to a pixel size of 10m to 50m. In that case however, we would also need to manually calculate NDVI etc., which adds a little bit of extra work.

ML Model	RMSE	$R^2$
KNeighborsRegressor	0.935147	0.679537
RandomForestRegressor	0.730378	0.804515
HistGradientBoostingRegressor	0.729455	0.805009
SVR	0.804774	0.762663

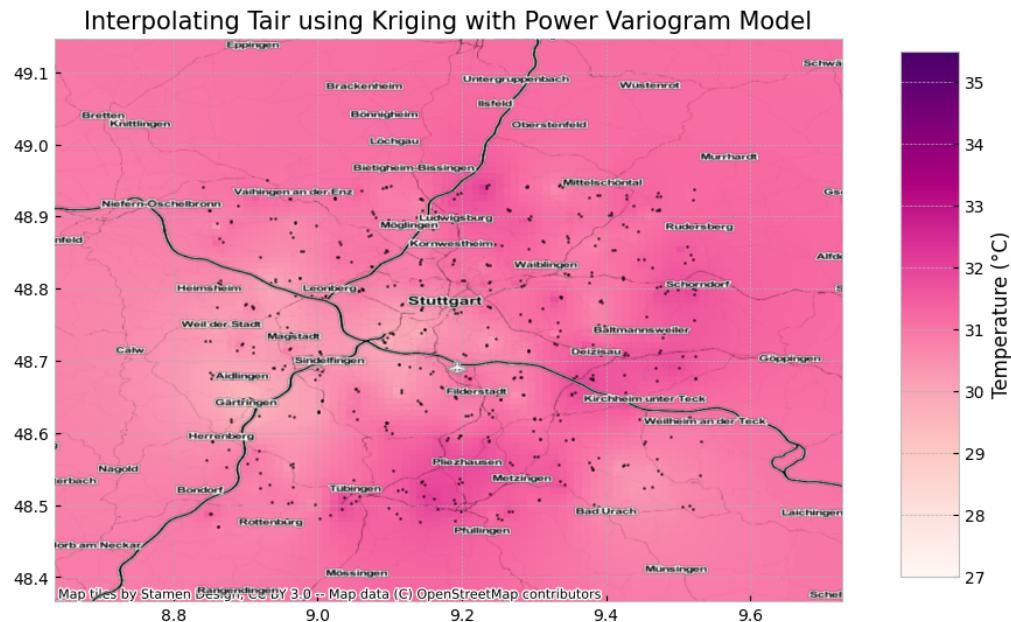


Figure 5.14: Ordinary Kriging Power Variogram, Netatmo Stuttgart, 19.06.2023 2PM

#### 5.2.4 Comparison with Netatmo

Like Sensor.Community, Netatmo also has many stations in the region of Stuttgart. To compare the two providers, we use Netatmo data for the same timestamp, i.e., 2PM on the 19. June 2023, to get a rough comparison. Please note, that we did not apply any QC to this data. This comparison should be improved by also applying QC steps to Netatmo stations; however, we did not have the time to collect this data and do the pre-processing steps in this work. The OK results are shown in Table 5.2.4. In this comparison, the power variogram model seems to perform the best. The resulting map can be found in

Figure 5.14. The next step would be to also run through all pre-processing steps with Netatmo data for QC etc. and also compare the ML models with each other on the Netatmo data, possibly also on a combination of Netatmo and Sensor.Community to get even higher spatial coverage.

Variogram Model	RMSE	R <sup>2</sup>
linear	1.755016	0.070271
power	1.592532	0.234456
gaussian	1.746884	0.078868
spherical	1.722252	0.104662
exponential	1.669650	0.158518

Table 5.3: RMSE and R<sup>2</sup> of Ordinary Kriging for Stuttgart 19.06.2023 2PM, Netatmo



## 6 Conclusion

In this work, the feasibility of two major use cases for ML-based interpolation, e.g., interpolation for a single location and areal interpolation, was explored given the context of urban local  $T_{air}$  mapping and an upper bound for RMSE values for both use cases were established, comparing different ML models against each other and testing their suitability to model the urban microclimate.

For interpolation of a single station, sensor readings from Netatmo and Sensor.Community were collected for the month of June 2023 from the areas of Hamburg and Stuttgart in Germany. The lowest RMSE was achieved by HistGB of around 0.5, depending on the train/test split, showing the ability of the ML model to accurately predict  $T_{air}$  for a specific location based on surrounding stations, even when stations have missing data and are not available at all times. HistGB was further evaluated for the region of Hamburg with Netatmo data to inspect feature (permutation) importance, distances between neighbours, and data availability of neighbours. A look at the permutation importance suggests, that only a small percentage of neighbouring stations, i.e., those that have a very similar temperature curve, have a huge influence on the prediction quality, while other neighbours and other features have little to no influence, suggesting that for single station interpolation one might only want to focus on collecting temperature data and ignore other types of readings, reducing data pre-processing overhead. The distance between neighbours had only a minimal influence up to 7km, suggesting that neighbouring stations should not be located further than 10km away. Experiments with sampling the available data per station to simulate non-stationary sensors showed that even with a big number of moving sensors, as long as there are few stationary sensors, prediction quality is always high; however, when those stationary sensors get removed and all sensors are moving, errors only stayed low when many readings, e.g., 1000 per station, were available, which could be achieved for example by a bus that visits a location more than once every hour ( $30\text{days} * 24\text{h} = 720$ ).

Areal interpolation on the other hand was evaluated with Sensor.Community data from Stuttgart and performed slightly worse than single station interpolation at a RMSE of 1.4-1.5 for a single timestamp on the 19.06.2023 14h, an especially hot day in Stuttgart, during the day but with a significantly lower  $R^2$  score of a maximum of 0.2 with some models. In comparison, on the same day but in the night at 5AM, the RMSE was significantly lower down to 0.65-0.7 for HistGB with  $R^2$  scores around 0.7, suggesting that during the day other factors influence the urban climate that we do not capture, such as LST or solar radiance, which should be explored in the future. Compared to OK, a geostatistical model,

the best predictor, i.e., HistGB, had similar RMSE and  $R^2$  scores when tested against both timestamps; however, the RMSE and  $R^2$  values highly depended on the way the data was split, as changing the random seed for the train/test splitting revealed, suggesting as there is information missing for the model to accurately evaluate specific stations and sensor readings. One such missing factor could be the height of sensors, as a comparison with a DWD reference station revealed, that the temperature at 2m and 5cm differs by a lot, which means that a hotter station that is closer to the ground as a cooler station nearby could be correct; however, in the model this information is not captured, resulting in training data where two totally different sensor readings are located next to each other. Finally, we have shown that HistGB seems to be a very flexible model that outperforms similar RF models, given that there is slightly more data to train on, and also has the added bonus of natively handling missing values, and is suitable for  $T_{air}$  interpolation in the urban setting. Due to the limited scope, there were many aspects which could be further improved such as using 10-fold cross validation everywhere and running exhaustive grid searches to find the most optimal model setups, collecting more data for more cities, eventually with better ways to validate our findings, i.e., through more densely located reference stations, improving the QC process, using more precise remote sensing data, and much more.

## 6.1 Future Outlook

There are many different directions in which this work could be extended. The areal interpolation use-case offers much room for improvement by using more precise remote sensing measurements or incorporating the temporal aspect, as well as the single station interpolation use-case, which could be turned into an actual study to collect data and evaluate how this data could improve the monitoring of the urban climate.

Additionally, we didn't compare ML models to Neural Networks, which could be even more powerful in their predictive capabilities. Another starting point could also be data pre-processing and QC or the influence of sensor height (and location) on the prediction quality. Lastly, working with spatial data purely in Python was quite slow, therefore it could make sense to improve tooling around the interpolation and PWS data collection to provide researchers with more flexible ways of conducting research and reducing barriers of getting started.

---

# Bibliography

- [Alt92] ALTMAN, Naomi S.: An introduction to kernel and nearest-neighbor non-parametric regression. In: *The American Statistician* 46 (1992), Nr. 3, S. 175–185
- [AR20] ALONSO, Lucille ; RENARD, Florent: A new approach for understanding urban microclimate by integrating complementary predictors at different scales in regression and machine learning models. In: *Remote Sensing* 12 (2020), Nr. 15, S. 2434
- [AYDK22] APAYDIN, Merve ; YUMUŞ, Mehmetan ; DEĞIRMENCI, Ali ; KARAL, Ömer: Evaluation of air temperature with machine learning regression methods using Seoul City meteorological data. In: *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi* 28 (2022), Nr. 5, S. 737–747
- [BJK<sup>+</sup>19] BORNHOLDT, Heiko ; JOST, David ; KISTERS, Philipp ; ROTTLEUTHNER, Michel ; BADE, Dirk ; LAMERSDORF, Winfried ; SCHMIDT, Thomas C. ; FISCHER, Mathias: SANE: Smart networks for urban citizen participation. In: *2019 26th International Conference on Telecommunications (ICT)* IEEE, 2019, S. 496–500
- [BP47] BALCHIN, William George V. ; PYE, Norman: A micro-climatological investigation of bath and the surrounding district. In: *Quarterly Journal of the Royal Meteorological Society* 73 (1947), Nr. 317-318, S. 297–323
- [Bre96] BREIMAN, Leo: Bagging predictors. In: *Machine learning* 24 (1996), S. 123–140
- [Bre99] BREIMAN, Leo: Pasting small votes for classification in large databases and on-line. In: *Machine learning* 36 (1999), S. 85–103
- [Bre01] BREIMAN, Leo: Random forests. In: *Machine learning* 45 (2001), S. 5–32
- [Büc18] BÜCHAU, Yann G.: *Modelling Shield Temperature Sensors: An Assessment of the Netatmo Citizen Weather Station*, Universität Hamburg, Diss., 2018
- [CDS<sup>+</sup>17] CASTELL, Nuria ; DAUGE, Franck R. ; SCHNEIDER, Philipp ; VOGL, Matthias ; LERNER, Uri ; FISHBAIN, Barak ; BRODAY, David ;

- BARTONOVA, Alena: Can commercial low-cost sensor platforms contribute to air quality monitoring and exposure estimates? In: *Environment international* 99 (2017), S. 293–302
- [CH65] CHORLEY, Richard J. ; HAGGETT, Peter: Trend-surface mapping in geographical research. In: *Transactions of the Institute of British Geographers* (1965), S. 47–67
- [CH67] COVER, Thomas ; HART, Peter: Nearest neighbor pattern classification. In: *IEEE transactions on information theory* 13 (1967), Nr. 1, S. 21–27
- [CK03] CHONG, Chee-Yee ; KUMAR, Srikanta P.: Sensor networks: evolution, opportunities, and challenges. In: *Proceedings of the IEEE* 91 (2003), Nr. 8, S. 1247–1256
- [CMY<sup>+</sup>15] CHAPMAN, Lee ; MULLER, Catherine L. ; YOUNG, Duick T. ; WARREN, Elliott L. ; GRIMMOND, C Sue B. ; CAI, Xiao-Ming ; FERRANTI, Emma J.: The Birmingham urban climate laboratory: an open meteorological test bed and challenges of the smart city. In: *Bulletin of the American Meteorological Society* 96 (2015), Nr. 9, S. 1545–1560
- [cop] *ESA - Copernicus DEM GLO-30 dataset*. <https://doi.org/10.5270/esa-c5d3d65>
- [CW11] CAMPBELL, James B. ; WYNNE, Randolph H.: *Introduction to remote sensing*. Guilford Press, 2011
- [Did21] DIDAN, K: MODIS/Terra vegetation indices 16-day L3 global 1 km SIN grid V061 [data set]. In: *NASA EOSDIS Land Processes DAAC*. Available online: <https://lpdaac.usgs.gov/products/mod13a2v061/> (accessed on 1 August 2023) (2021). <http://dx.doi.org/10.5067/MODIS/MOD13A2.061>. – DOI 10.5067/MODIS/MOD13A2.061
- [DP10] DARGIE, Waltenegus ; POELLABAUER, Christian: *Fundamentals of wireless sensor networks: theory and practice*. John Wiley & Sons, 2010
- [DRTP19] DIRKSEN, M ; RONDA, RJ ; THEEUWES, NE ; PAGANI, GA: Sky view factor calculations and its application in urban heat island studies. In: *Urban Climate* 30 (2019), S. 100498
- [FBD<sup>+</sup>21] FENNER, Daniel ; BECHTEL, Benjamin ; DEMUZERE, Matthias ; KITTNER, Jonas ; MEIER, Fred: CrowdQC+—a quality-control for crowdsourced air-temperature observations enabling world-wide urban climate applications. In: *Frontiers in Environmental Science* 9 (2021), S. 553

- [FHM<sup>+</sup>19] FENNER, Daniel ; HOLTMANN, Achim ; MEIER, Fred ; LANGER, Ines ; SCHERER, Dieter: Contrasting changes of urban heat island intensity during hot weather episodes. In: *Environmental Research Letters* 14 (2019), Nr. 12, S. 124013
- [Gea54] GEARY, Robert C.: The contiguity ratio and statistical mapping. In: *The incorporated statistician* 5 (1954), Nr. 3, S. 115–146
- [GHD<sup>+</sup>17] GORELICK, Noel ; HANCHER, Matt ; DIXON, Mike ; ILYUSHCHENKO, Simon ; THAU, David ; MOORE, Rebecca: Google Earth Engine: Planetary-scale geospatial analysis for everyone. In: *Remote sensing of Environment* 202 (2017), S. 18–27
- [Goo16] GOOD, Elizabeth J.: An in situ-based analysis of the relationship between land surface “skin” and screen-level air temperatures. In: *Journal of Geophysical Research: Atmospheres* 121 (2016), Nr. 15, S. 8801–8819
- [GRGTDW20] GRÊT-REGAMEY, Adrienne ; GALLEGUILLOS-TORRES, Marcelo ; DISSEGNA, Angela ; WEIBEL, Bettina: How urban densification influences ecosystem services—a comparison between a temperate and a tropical city. In: *Environmental Research Letters* 15 (2020), Nr. 7, S. 075001
- [Gri06] GRIMMOND, CSB: Progress in measuring and observing the urban atmosphere. In: *Theoretical and Applied Climatology* 84 (2006), S. 3–22
- [GVP] GHENT, D. ; VEAL, K. ; PERRY, M.: *ESA Land Surface Temperature Climate Change Initiative (LST\_cci): Multisensor Infra-Red (IR) Low Earth Orbit (LEO) land surface temperature (LST) time series level 3 supercollated (L3S) global product (1995-2020), version 2.00.* <http://dx.doi.org/10.5285/ef8ce37b6af24469a2a4bdc31d3db27d>
- [Hai01] HAINING, RP: Spatial autocorrelation. (2001). <http://dx.doi.org/10.1016/B0-08-043076-7/02511-0>. – DOI 10.1016/B0-08-043076-7/02511-0
- [HDJ<sup>+</sup>02] HARVEY, Nicholas J. ; DUNAGAN, John ; JONES, Mike ; SAROIU, Stefan ; THEIMER, Marvin ; WOLMAN, Alec: Skipnet: A scalable overlay network with practical locality properties. (2002)
- [HGMS<sup>+</sup>22] HAHN, Claudia ; GARCIA-MARTI, Irene ; SUGIER, Jacqueline ; EMSLEY, Fiona ; BEAULANT, Anne-Lise ; ORAM, Louise ; STRANDBERG, Eva ; LINDGREN, Elisa ; SUNTER, Martyn ; ZISKA, Franziska: Observations from Personal Weather Stations—EUMETNET Interests and Experience. In: *Climate* 10 (2022), Nr. 12, S. 192

- [HKS<sup>+</sup>14] HO, Hung C. ; KNUDBY, Anders ; SIROVYAK, Paul ; XU, Yongming ; HODUL, Matus ; HENDERSON, Sarah B.: Mapping maximum urban air temperature on hot summer days. In: *Remote Sensing of Environment* 154 (2014), S. 38–45
- [HNW<sup>+</sup>18] HENGL, Tomislav ; NUSSBAUM, Madlene ; WRIGHT, Marvin N. ; HEUVELINK, Gerard B. ; GRÄLER, Benedikt: Random forest as a generic framework for predictive modeling of spatial and spatio-temporal variables. In: *PeerJ* 6 (2018), S. e5518
- [Ho98] HO, Tin K.: The random subspace method for constructing decision forests. In: *IEEE transactions on pattern analysis and machine intelligence* 20 (1998), Nr. 8, S. 832–844
- [How33] HOWARD, Luke: *The climate of London: deduced from meteorological observations made in the metropolis and at various places around it.* Bd. 3. Harvey and Darton, J. and A. Arch, Longman, Hatchard, S. Highley [and] R. Hunter, 1833
- [HSW89] HORNIK, Kurt ; STINCHCOMBE, Maxwell ; WHITE, Halbert: Multilayer feedforward networks are universal approximators. In: *Neural networks* 2 (1989), Nr. 5, S. 359–366
- [IBA<sup>+</sup>22] IYER, Shiva R. ; BALASHANKAR, Ananth ; AEBERHARD, William H. ; BHATTACHARYYA, Sujoy ; RUSCONI, Giuditta ; JOSE, Lejo ; SOANS, Nita ; SUDARSHAN, Anant ; PANDE, Rohini ; SUBRAMANIAN, Lakshminarayanan: Modeling fine-grained spatio-temporal pollution maps with low-cost sensors. In: *npj Climate and Atmospheric Science* 5 (2022), Nr. 1, S. 76
- [iee19] IEEE Standard for Floating-Point Arithmetic. In: *IEEE Std 754-2019 (Revision of IEEE 754-2008)* (2019), S. 1–84. <http://dx.doi.org/10.1109/IEEESTD.2019.8766229>. – DOI 10.1109/IEEESTD.2019.8766229
- [K<sup>+</sup>95] KOHAVI, Ron u. a.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Ijcai* Bd. 14 Montreal, Canada, 1995, S. 1137–1145
- [KA06] KAMADA, Yuya ; ABE, Shigeo: Support vector regression using mahalanobis kernels. In: *Artificial Neural Networks in Pattern Recognition: Second IAPR Workshop, ANNPR 2006, Ulm, Germany, August 31-September 2, 2006. Proceedings* 2 Springer, 2006, S. 144–152
- [KB21] KIM, Se W. ; BROWN, Robert D.: Urban heat island (UHI) intensity and magnitude estimations: A systematic literature review. In: *Science of the Total Environment* 779 (2021), S. 146389

- [KBF<sup>+</sup>16] KJELLSTROM, Tord ; BRIGGS, David ; FREYBERG, Chris ; LEMKE, Bruno ; OTTO, Matthias ; HYATT, Olivia: Heat, human performance, and occupational health: a key issue for the assessment of global climate change impacts. In: *Annual review of public health* 37 (2016), S. 97–112
- [KH08] KOVATS, R S. ; HAJAT, Shakoor: Heat stress and public health: a critical review. In: *Annu. Rev. Public Health* 29 (2008), S. 41–55
- [KMF<sup>+</sup>17] KE, Guolin ; MENG, Qi ; FINLEY, Thomas ; WANG, Taifeng ; CHEN, Wei ; MA, Weidong ; YE, Qiwei ; LIU, Tie-Yan: Lightgbm: A highly efficient gradient boosting decision tree. In: *Advances in neural information processing systems* 30 (2017)
- [KPS<sup>+</sup>11] KOSKINEN, Jarkko T. ; POUTIAINEN, Jani ; SCHULTZ, David M. ; JOFFRE, Sylvain ; KOISTINEN, Jarmo ; SALTIKOFF, Elena ; GREGOW, Erik ; TURTIAINEN, Heikki ; DABBERDT, Walter F. ; DAMSKI, Juhani u. a.: The Helsinki Testbed: a mesoscale measurement, research, and service platform. In: *Bulletin of the American Meteorological Society* 92 (2011), Nr. 3, S. 325–342
- [Kri76] KRIGE, DG: A review of the development of geostatistics in South Africa. In: *Advanced Geostatistics in the Mining Industry: Proceedings of the NATO Advanced Study Institute held at the Istituto di Geologia Applicata of the University of Rome, Italy, 13–25 October 1975* Springer, 1976, S. 279–293
- [Lam83] LAM, Nina Siu-Ngan: Spatial interpolation methods: a review. In: *The American Cartographer* 10 (1983), Nr. 2, S. 129–150
- [LBH15] LECUN, Yann ; BENGIO, Yoshua ; HINTON, Geoffrey: Deep learning. In: *nature* 521 (2015), Nr. 7553, S. 436–444
- [LF14] LECLERC, Monique Y. ; FOKEN, Thomas: *Footprints in micrometeorology and ecology*. Bd. 239. Springer, 2014
- [LG12] LOUPPE, Gilles ; GEURTS, Pierre: Ensembles on random patches. In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, September 24–28, 2012. Proceedings, Part I* 23 Springer, 2012, S. 346–361
- [LH14] LI, Jin ; HEAP, Andrew D.: Spatial interpolation methods applied in the environmental sciences: A review. In: *Environmental Modelling & Software* 53 (2014), S. 173–189
- [Low77] LOWRY, William P.: Empirical estimation of urban effects on climate: a problem analysis. In: *Journal of Applied Meteorology and Climatology* 16 (1977), Nr. 2, S. 129–135

- [LPS18] LEWIS, Alastair ; PELTIER, W R. ; SCHNEIDEMESSER, Erika von: Low-cost sensors for the measurement of atmospheric composition: overview of topic and future applications. (2018)
- [LSF19] LORENZ, Ruth ; STALHANDSKE, Zélie ; FISCHER, Erich M.: Detection of a climate change signal in extreme heat, heat stress, and cold in Europe from observations. In: *Geophysical Research Letters* 46 (2019), Nr. 14, S. 8363–8374
- [MBG15] MARTIN, Philippe ; BAUDOUIN, Yves ; GACHON, Philippe: An alternative method to characterize the surface urban heat island. In: *International journal of biometeorology* 59 (2015), S. 849–861
- [MCG<sup>+</sup>13] MULLER, Catherine L. ; CHAPMAN, Lee ; GRIMMOND, CSB ; YOUNG, Duick T. ; CAI, Xiaoming: Sensors and the city: a review of urban meteorological networks. In: *International Journal of Climatology* 33 (2013), Nr. 7, S. 1585–1600
- [MFG<sup>+</sup>17] MEIER, Fred ; FENNER, Daniel ; GRASSMANN, Tom ; OTTO, Marco ; SCHERER, Dieter: Crowdsourcing air temperature from citizen weather stations for urban climate research. In: *Urban Climate* 19 (2017), S. 170–191
- [MKP21] MYNENI, R ; KNYAZIKHIN, Y ; PARK, T: MODIS/Terra Leaf Area Index/FPAR 8-Day L4 Global 500m SIN Grid V061 [Dataset]. NASA EOSDIS Land Processes DAAC / Accessed 2023-08-09 from <https://doi.org/10.5067/MODIS/MOD15A2H.061>. 2021. – Forschungsbericht
- [MM88] MITÁŠ, L ; MITÁŠOVÁ, H: General variational approach to the interpolation problem. In: *Computers & Mathematics with Applications* 16 (1988), Nr. 12, S. 983–992
- [Mor48] MORAN, Patrick A.: The interpretation of statistical maps. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 10 (1948), Nr. 2, S. 243–251
- [MPV21] MONTGOMERY, Douglas C. ; PECK, Elizabeth A. ; VINING, G G.: *Introduction to linear regression analysis*. John Wiley & Sons, 2021
- [MR06] MEINSHAUSEN, Nicolai ; RIDGEWAY, Greg: Quantile regression forests. In: *Journal of machine learning research* 7 (2006), Nr. 6
- [MYM22] MURPHY, Benjamin ; YURCHAK, Roman ; MÜLLER, Sebastian: *GeoStat-Framework/PyKriging*: v1.7.0. <http://dx.doi.org/10.5281/zenodo.7008206>. Version: August 2022

- [NAEB23] NJOKU, Elijah A. ; AKPAN, Patrick E. ; EFFIONG, Augustine E. ; BABATUNDE, Isaac O.: The effects of station density in geostatistical prediction of air temperatures in Sweden: A comparison of two interpolation techniques. In: *Resources, Environment and Sustainability* 11 (2023), S. 100092
- [Oke73] OKE, Tim R.: City size and the urban heat island. In: *Atmospheric Environment* (1967) 7 (1973), Nr. 8, S. 769–779
- [Oke76] OKE, Timothy R.: The distinction between canopy and boundary-layer urban heat islands. In: *Atmosphere* 14 (1976), Nr. 4, S. 268–277
- [Oke04] OKE, TR: *Siting and exposure of meteorological instruments at urban sites. 27th NATO/CCMS Int Tech Meeting on Air Pollution Modelling and Application.* 2004
- [Oke06] OKE, Timothy R.: Initial guidance to obtain representative meteorological observations at urban sites. (2006), 01, S. 51
- [OMCV17] OKE, Timothy R. ; MILLS, Gerald ; CHRISTEN, Andreas ; VOOGT, James A.: *Urban climates.* Cambridge University Press, 2017
- [OMMF17] OBRADOVICH, Nick ; MIGLIORINI, Robyn ; MEDNICK, Sara C. ; FOWLER, James H.: Nighttime temperature and human sleep loss in a changing climate. In: *Science advances* 3 (2017), Nr. 5, S. e1601555
- [OMPR18] OBRADOVICH, Nick ; MIGLIORINI, Robyn ; PAULUS, Martin P. ; RAHWAN, Iyad: Empirical evidence of mental health risks posed by climate change. In: *Proceedings of the National Academy of Sciences* 115 (2018), Nr. 43, S. 10953–10958
- [Ope23] OPENAI: *GPT-4 Technical Report.* 2023
- [PPC<sup>+</sup>12] PENG, Shushi ; PIAO, Shilong ; CIAIS, Philippe ; FRIEDLINGSTEIN, Pierre ; OTTLE, Catherine ; BRÉON, François-Marie ; NAN, Huijuan ; ZHOU, Liming ; MYNENI, Ranga B.: Surface urban heat island across 419 global big cities. In: *Environmental science & technology* 46 (2012), Nr. 2, S. 696–703
- [PVG<sup>+</sup>11] PEDREGOSA, F. ; VAROQUAUX, G. ; GRAMFORT, A. ; MICHEL, V. ; THIRION, B. ; GRISEL, O. ; BLONDEL, M. ; PRETTENHOFER, P. ; WEISS, R. ; DUBOURG, V. ; VANDERPLAS, J. ; PASSOS, A. ; COURNAPEAU, D. ; BRUCHER, M. ; PERROT, M. ; DUCHESNAY, E.: Scikit-learn: Machine Learning in Python. In: *Journal of Machine Learning Research* 12 (2011), S. 2825–2830

- [RGA<sup>+</sup>09] RUNDEL, Philip W. ; GRAHAM, Eric A. ; ALLEN, Michael F. ; FISHER, Jason C. ; HARMON, Thomas C.: Environmental sensor networks in ecological research. In: *New Phytologist* 182 (2009), Nr. 3, S. 589–607
- [RO00] RUNNALLS, KE ; OKE, TR: Dynamics and controls of the near-surface heat island of Vancouver, British Columbia. In: *Physical Geography* 21 (2000), Nr. 4, S. 283–304
- [Ros57] ROSENBLATT, Frank: *The perceptron, a perceiving and recognizing automaton Project Para.* Cornell Aeronautical Laboratory, 1957
- [Sar21] SARKER, Iqbal H.: Machine learning: Algorithms, real-world applications and research directions. In: *SN computer science* 2 (2021), Nr. 3, S. 160
- [SB92] STOLL, Matthew J. ; BRAZEL, Anthony J.: Surface-air temperature relationships in the urban environment of Phoenix, Arizona. In: *Physical Geography* 13 (1992), Nr. 2, S. 160–179
- [SBZH07] STROBL, Carolin ; BOULESTEIX, Anne-Laure ; ZEILEIS, Achim ; HOTHORN, Torsten: Bias in random forest variable importance measures: Illustrations, sources and a solution. In: *BMC bioinformatics* 8 (2007), Nr. 1, S. 1–21
- [sen] *Sentinel-2 MSI: MultiSpectral Instrument, Level-2A.* [https://developers.google.com/earth-engine/datasets/catalog/COPERNICUS\\_S2\\_SR](https://developers.google.com/earth-engine/datasets/catalog/COPERNICUS_S2_SR),
- [SHN<sup>+</sup>08] SUGAWARA, Hirofumi ; HAGISHIMA, Aya ; NARITA, Ken-ichi ; OGAWA, Hiroko ; YAMANO, Mitsuo: Temperature and wind distribution in an EW-oriented urban street canyon. In: *SOLA* 4 (2008), S. 53–56
- [SJVL<sup>+</sup>13] STONE JR, Brian ; VARGO, Jason ; LIU, Peng ; HU, Yongtao ; RUSSELL, Armistead: Climate change adaptation through urban heat management in Atlanta, Georgia. In: *Environmental science & technology* 47 (2013), Nr. 14, S. 7780–7786
- [SKH18] SILVA, Bhagya N. ; KHAN, Murad ; HAN, Kijun: Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities. In: *Sustainable cities and society* 38 (2018), S. 697–713
- [SKH<sup>+</sup>20] SEKULIĆ, Aleksandar ; KLIBARDA, Milan ; HEUVELINK, Gerard B. ; NIKOLIĆ, Mladen ; BAJAT, Branislav: Random forest spatial interpolation. In: *Remote Sensing* 12 (2020), Nr. 10, S. 1687

- [SKP<sup>+</sup>20] SEKULIĆ, Aleksandar ; KLIBARDA, Milan ; PROTIĆ, Dragutin ; TADIĆ, Melita P. ; BAJAT, Branislav: Spatio-temporal regression kriging model of mean daily temperature for Croatia. In: *Theoretical and Applied Climatology* 140 (2020), S. 101–114
- [SO09] STEWART, Iain ; OKE, TR: Newly developed “thermal climate zones” for defining and measuring urban heat island magnitude in the canopy layer. In: *Eighth Symposium on Urban Environment, Phoenix, AZ*, 2009
- [SO12] STEWART, Ian D. ; OKE, Tim R.: Local climate zones for urban temperature studies. In: *Bulletin of the American Meteorological Society* 93 (2012), Nr. 12, S. 1879–1900
- [Ste27] STEFFENSEN, Johan F.: *Interpolation*. Williams & Wilkins, 1927
- [Ste11] STEWART, Iain D.: A systematic review and scientific critique of methodology in modern urban heat island literature. In: *International Journal of Climatology* 31 (2011), Nr. 2, S. 200–217
- [Sun51] SUNDBORG, Åke: *Climatological studies in Uppsala: With special regard to the temperature conditions in the urban area*. 1951
- [TDFH<sup>+</sup>22] THOPPILAN, Romal ; DE FREITAS, Daniel ; HALL, Jamie ; SHAZER, Noam ; KULSHRESHTHA, Apoorv ; CHENG, Heng-Tze ; JIN, Alicia ; BOS, Taylor ; BAKER, Leslie ; DU, Yu u. a.: Lamda: Language models for dialog applications. In: *arXiv preprint arXiv:2201.08239* (2022)
- [TMJ10] TAI, Amos P. ; MICKLEY, Loretta J. ; JACOB, Daniel J.: Correlations between fine particulate matter (PM2. 5) and meteorological variables in the United States: Implications for the sensitivity of PM2. 5 to climate change. In: *Atmospheric environment* 44 (2010), Nr. 32, S. 3976–3984
- [TYU86] TRANGMAR, Bruce B. ; YOST, Russel S. ; UEHARA, Goro: Application of geostatistics to spatial studies of soil properties. In: *Advances in agronomy* 38 (1986), S. 45–94
- [UATMG20] ULLAH, Zaib ; AL-TURJMAN, Fadi ; MOSTARDA, Leonardo ; GAGLIARDI, Roberto: Applications of artificial intelligence and machine learning in smart cities. In: *Computer Communications* 154 (2020), S. 313–323
- [UNSA19] UNITED NATIONS, Department of E. ; SOCIAL AFFAIRS, Population D.: *World Urbanization Prospects: The 2018 Revision*. <https://population.un.org/wup/publications/Files/WUP2018-Report.pdf>, 2019

- [VBEM20] VENTER, Zander S. ; BROUSSE, Oscar ; ESAU, Igor ; MEIER, Fred: Hyper-local mapping of urban air temperature using remote sensing and crowd-sourced weather data. In: *Remote Sensing of Environment* 242 (2020), S. 111791
- [VO03] VOOGT, James A. ; OKE, Tim R.: Thermal remote sensing of urban climates. In: *Remote sensing of environment* 86 (2003), Nr. 3, S. 370–384
- [VS17] VOELKEL, Jackson ; SHANDAS, Vivek: Towards systematic prediction of urban heat islands: Grounding measurements, assessing modeling techniques. In: *Climate* 5 (2017), Nr. 2, S. 41
- [VSZ02] VON STORCH, Hans ; ZWIERS, Francis W.: *Statistical analysis in climate research*. Cambridge university press, 2002
- [Wac03] WACKERNAGEL, Hans: *Multivariate geostatistics: an introduction with applications*. Springer Science & Business Media, 2003
- [WMO18] WMO, WMO: Guide to instruments and methods of observation. In: *World Meteorological Organization WMO*, URL <https://library.wmo.int/index.php> (2018)
- [WRP85] WILLMOTT, Cort J. ; ROWE, Clinton M. ; PHILPOT, William D.: Small-scale climate maps: A sensitivity analysis of some common assumptions associated with grid-point interpolation and contouring. In: *The American Cartographer* 12 (1985), Nr. 1, S. 5–16
- [WYC<sup>+</sup>16] WARREN, Elliott L. ; YOUNG, Duick T. ; CHAPMAN, Lee ; MULLER, Catherine ; GRIMMOND, CSB ; CAI, Xiao-Ming: The Birmingham Urban Climate Laboratory—A high density, urban meteorological dataset, from 2012–2014. In: *Scientific data* 3 (2016), Nr. 1, S. 1–8
- [Wyn85] WYNGAARD, John C.: Structure of the planetary boundary layer and implications for its modeling. In: *Journal of Applied Meteorology and Climatology* 24 (1985), Nr. 11, S. 1131–1142
- [YBZ19] YANG, Jiachuan ; BOU-ZEID, Elie: Designing sensor networks to resolve spatio-temporal urban temperature variations: fixed, mobile or hybrid? In: *Environmental Research Letters* 14 (2019), Nr. 7, S. 074022
- [YSL<sup>+</sup>20] YUAN, Qiangqiang ; SHEN, Huanfeng ; LI, Tongwen ; LI, Zhiwei ; LI, Shuwen ; JIANG, Yun ; XU, Hongzhang ; TAN, Weiwei ; YANG, Qian-qian ; WANG, Jiwen u. a.: Deep learning in environmental remote sensing: Achievements and challenges. In: *Remote Sensing of Environment* 241 (2020), S. 111716

- [ZIE10] ZUUR, Alain F. ; IENO, Elena N. ; ELPHICK, Chris S.: A protocol for data exploration to avoid common statistical problems. In: *Methods in ecology and evolution* 1 (2010), Nr. 1, S. 3–14
- [ZKBK21] ZUMWALD, Marius ; KNÜSEL, Benedikt ; BRESCH, David N. ; KNUTTI, Reto: Mapping urban temperature using crowd-sensing data and machine learning. In: *Urban Climate* 35 (2021), S. 100739
- [ZL12] ZHANG, Guoyi ; LU, Yan: Bias-corrected random forests in regression. In: *Journal of Applied Statistics* 39 (2012), Nr. 1, S. 151–160
- [ZPL15] ZHANG, Xiaoyu ; PANG, Jing ; LI, Lingling: Estimation of land surface temperature under cloudy skies using combined diurnal solar radiation and surface temperature evolution. In: *Remote Sensing* 7 (2015), Nr. 1, S. 905–921



# Appendix

## 1 Netatmo

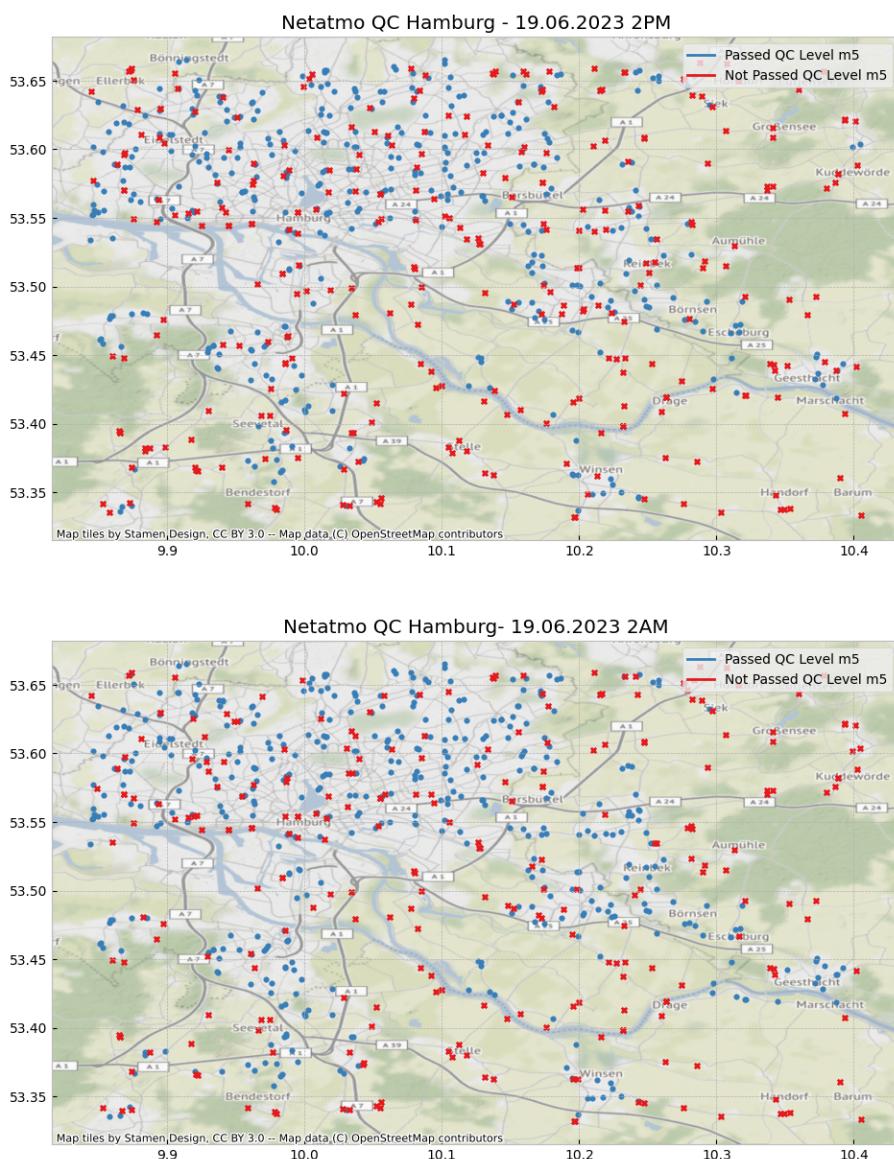


Figure 1: QC for Netatmo in Hamburg showing difference between day and night, 19.06.2023

## 2 Sklearn Model Parameters for Single Station Interpolation

Listing 1: Random Forest Regressor Parameters

```

1 class sklearn.ensemble.RandomForestRegressor(n_estimators=200,
2 *, criterion='squared_error', max_depth=None, min_samples_split=2,
3 min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=1.0,
4 max_leaf_nodes=None, min_impurity_decrease=0.0, bootstrap=True,
5 oob_score=False, n_jobs=None, random_state=None, verbose=0,
6 warm_start=False, ccp_alpha=0.0, max_samples=None)

```

The number of trees was increased from 100 to 200 to increase the performance.

Listing 2: Histogram-based Gradient Boosting Parameters

```

1 class sklearn.ensemble.HistGradientBoostingRegressor
2 (loss='squared_error', *, quantile=None, learning_rate=0.1,
3 max_iter=200, max_leaf_nodes=31, max_depth=None, min_samples_leaf=20,
4 l2_regularization=0.0, max_bins=255, categorical_features=None,
5 monotonic_cst=None, interaction_cst=None, warm_start=False,
6 early_stopping='auto', scoring='loss', validation_fraction=0.1,
7 n_iter_no_change=10, tol=1e-07, verbose=0, random_state=42)

```

The number of max iterations was increased from the default value of 100 to 200 to improve the performance of the model and the random state was set to 42 so all iterations yield the same result.

## 3 Feature Engineering

Listing 3: Timestamp to Sinus Curve Feature

```

1 # Convert datetime64 timestamp hours and minutes to circle angle
2 filtered_gdf['time'] = filtered_gdf['time'].apply(lambda x: (x.hour * 60 + x.minute) / 360 * 2 * np.pi)
3 filtered_gdf.rename(columns={'time': 'time_angle'}, inplace=True)
4
5 # Calculate continuous representation of time as an angle in radians
6 filtered_gdf['sin_time'] = np.sin(filtered_gdf['time_angle'])

```

## 4 Histogram-based Gradient Boosting Single Location Interpolation

### List of stations for minimum distance between stations

The list of stations for the minimum distance between stations is the following by their Netatmo station id:

- 70:ee:50:83:b1:e2
- 70:ee:50:16:16:ce
- 70:ee:50:5e:d4:16
- 70:ee:50:00:d3:96
- 70:ee:50:6b:5f:50
- 70:ee:50:00:d1:1c
- 70:ee:50:5f:51:a0
- 70:ee:50:6b:97:86
- 70:ee:50:28:f2:ca
- 70:ee:50:58:e8:70

The locations of the stations are presented in Figure 4.

## 5 Areal Interpolation

Figure 5 shows the mean/max/min  $T_{air}$  for DWD station with id 4931 across June 2023 in Stuttgart near the airport. Figure 6 shows the  $T_{air}$  for the same station on the 19. June 2023 in a 10-minute interval. The blue line, i.e., TT\_10, shows the  $T_{air}$  at 2m while the orange line, i.e., TM5\_10, shows the  $T_{air}$  at 5cm directly above the ground.

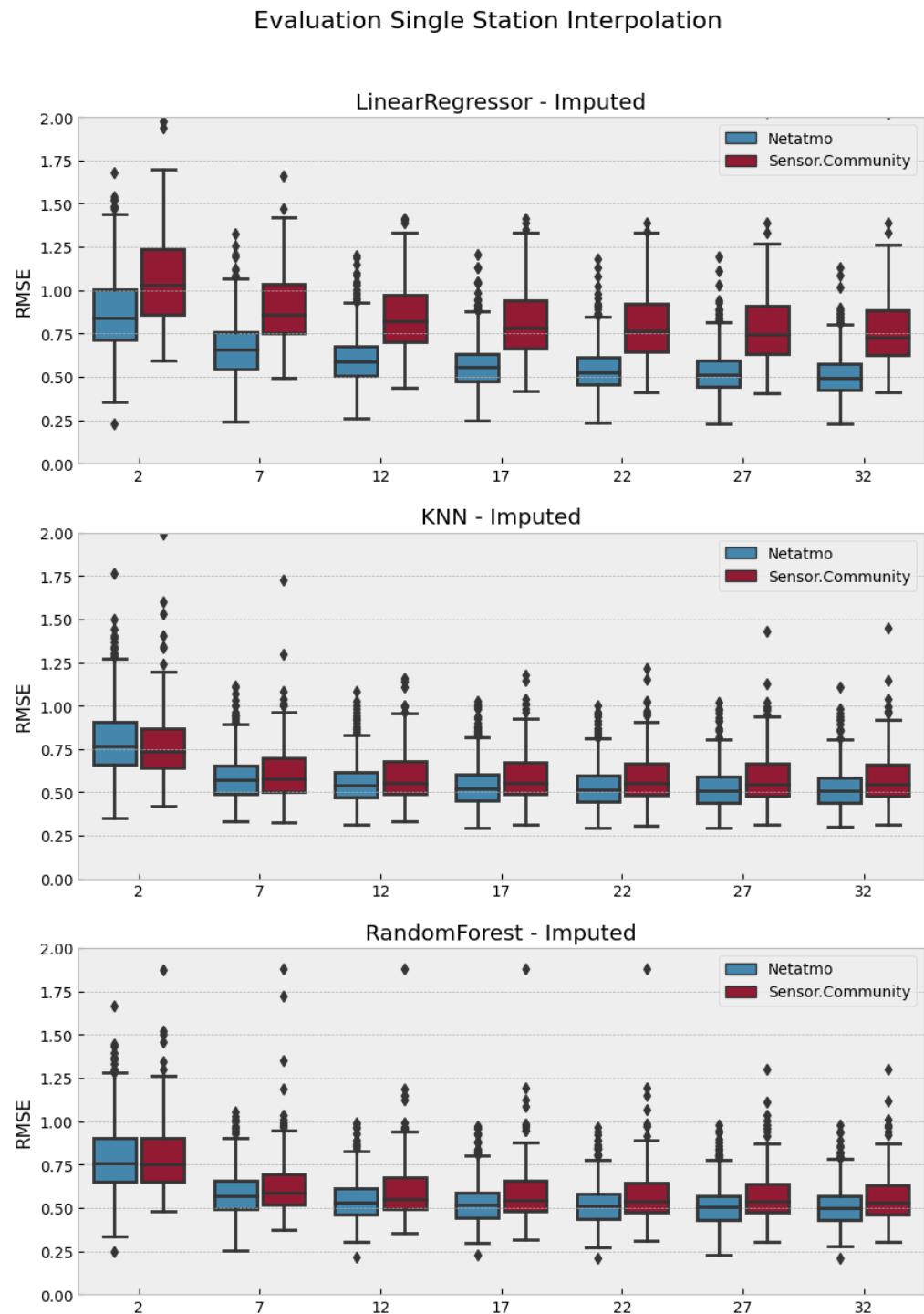


Figure 2: Evaluation Single Station Interpolation, Detailed View

### Evaluation Single Station Interpolation Part 2

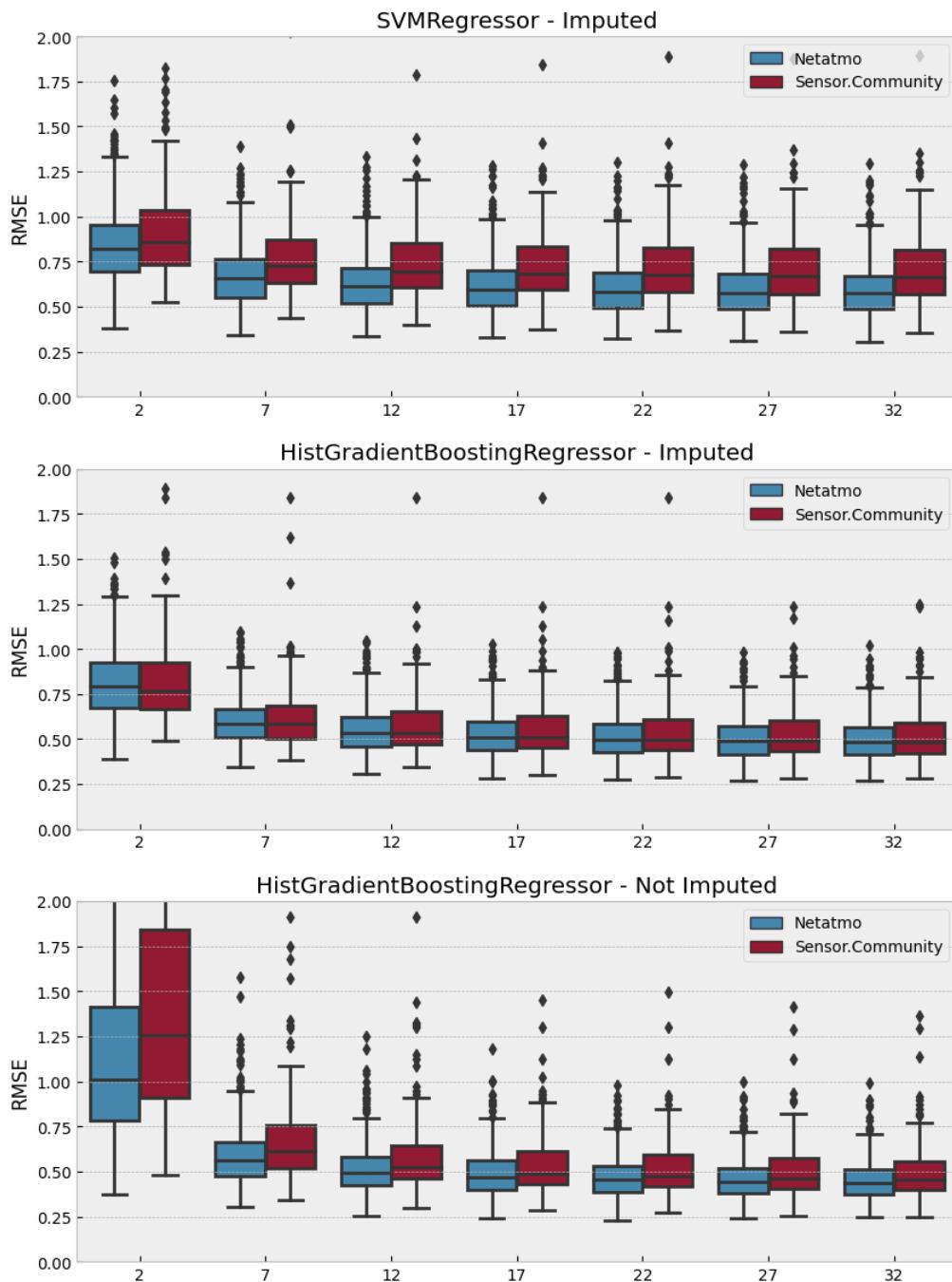


Figure 3: Evaluation Single Station Interpolation, Detailed View

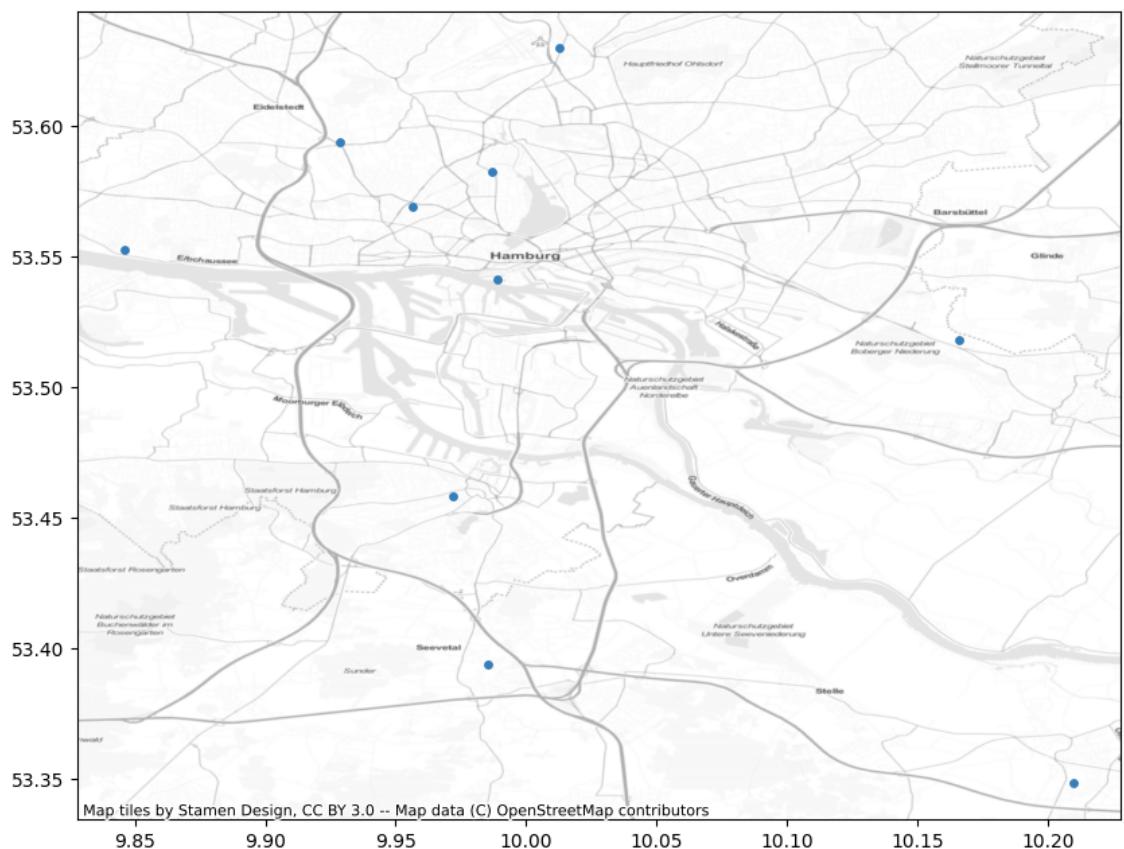


Figure 4: Netatmo Stations for Minimum Distance Between Stations, Hamburg

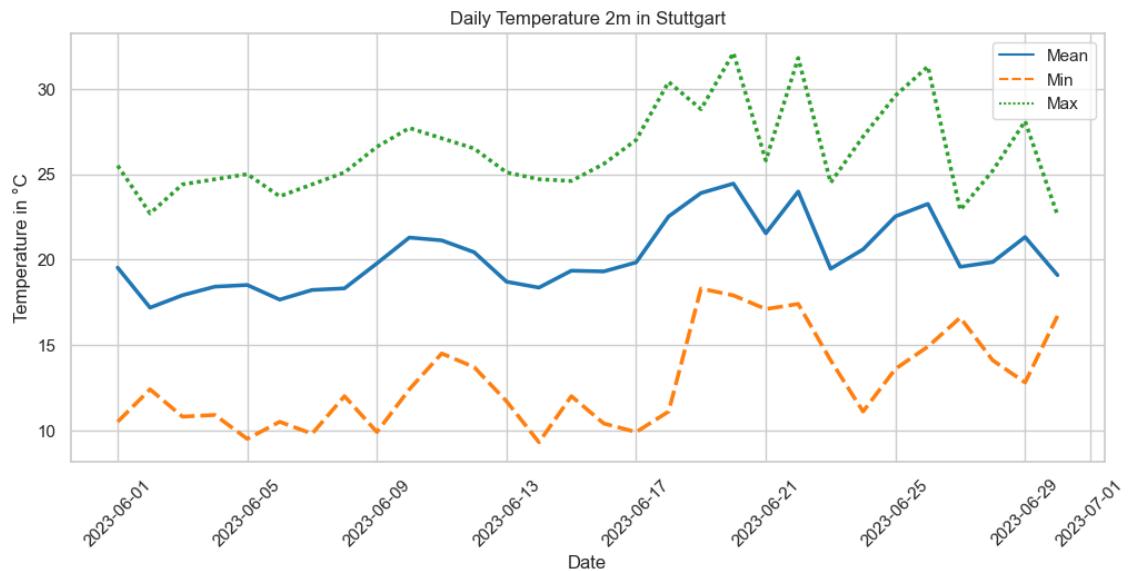


Figure 5: DWD Station 4931, Stuttgart, June 2023

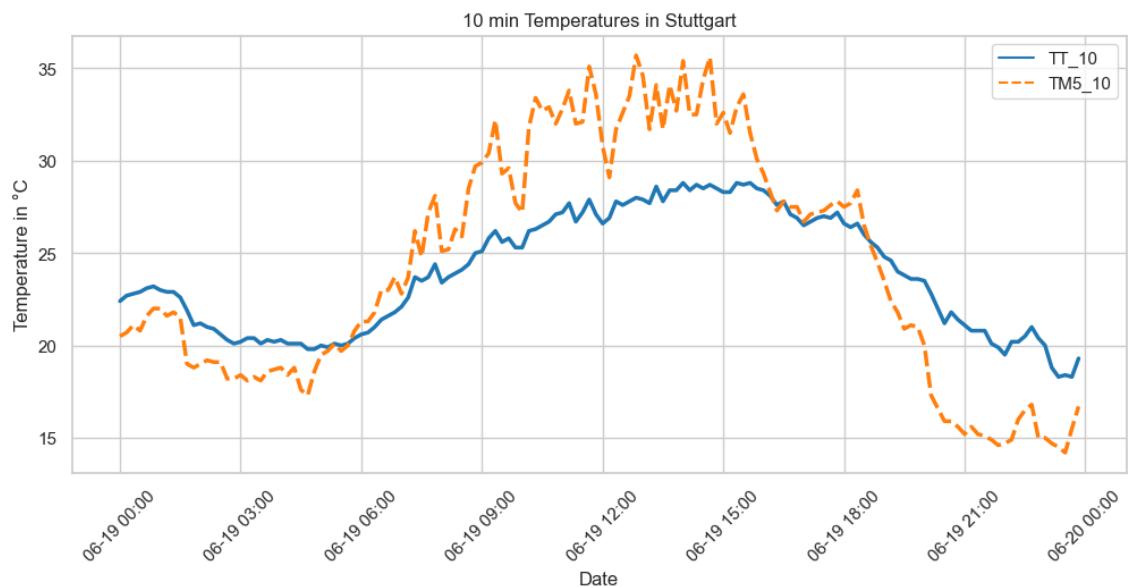


Figure 6: DWD Station 4931, Stuttgart, 19. June 2023

## 6 SensorCommunity

Listing 4: Sensor Distribution by Country

```
1  {
2      "2023-06-24": {
3          "bme280": {
4              "WORLD": "5006",
5              "DE": "1558"
6          },
7          "bmp180": {
8              "WORLD": "159",
9              "DE": "72"
10         },
11         "bmp280": {
12             "WORLD": "254",
13             "DE": "100"
14         },
15         "dht22": {
16             "WORLD": "5292",
17             "DE": "2590"
18         },
19         "ds18b20": {
20             "WORLD": "29",
21             "DE": "11"
22         },
23         "hpm": {
24             "WORLD": "6",
25             "DE": "1"
26         },
27         "htu21d": {
28             "WORLD": "105",
29             "DE": "14"
30         },
31         "laerm": {
32             "WORLD": "233",
33             "DE": "117"
34         },
35         "nextpm": {
36             "WORLD": "1",
37             "DE": "1"
38         },
39         "pms1003": {
40             "WORLD": "7",
41             "DE": "2"
42     },
```

```
43     "pms3003": {
44         "WORLD": "7"
45     },
46     "pms5003": {
47         "WORLD": "255",
48         "DE": "16"
49     },
50     "pms6003": {
51         "WORLD": "1",
52         "DE": "1"
53     },
54     "pms7003": {
55         "WORLD": "206",
56         "DE": "8"
57     },
58     "ppd42ns": {
59         "WORLD": "2",
60         "DE": "1"
61     },
62     "radiation_sbm-19": {
63         "WORLD": "3"
64     },
65     "radiation_sbm-20": {
66         "WORLD": "7"
67     },
68     "radiation_si22g": {
69         "WORLD": "71",
70         "DE": "54"
71     },
72     "scd30": {
73         "WORLD": "2"
74     },
75     "sds011": {
76         "WORLD": "12199",
77         "DE": "4964"
78     },
79     "sht15": {
80         "WORLD": "1",
81         "DE": "1"
82     },
83     "sht30": {
84         "WORLD": "130",
85         "DE": "17"
86     },
87     "sht31": {
88         "WORLD": "259",
```

```
89         "DE": "57"
90     },
91     "sht35": {
92         "WORLD": "11",
93         "DE": "5"
94     },
95     "sht85": {
96         "WORLD": "2",
97         "DE": "2"
98     },
99     "sps30": {
100        "WORLD": "279",
101        "DE": "53"
102    }
103}
104}
```

## **Eidesstattliche Versicherung**

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit im Masterstudien-  
gang Wirtschaftsinformatik selbstständig verfasst und keine anderen als die angegebe-  
nen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-  
Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichen-  
gen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin,  
dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe.

Hamburg, den \_\_\_\_\_ Unterschrift: \_\_\_\_\_

## **Veröffentlichung**

Ich stimme der Einstellung der Arbeit in die Bibliothek des Fachbereichs Informatik  
zu.

Hamburg, den \_\_\_\_\_ Unterschrift: \_\_\_\_\_