Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

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 continiously 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 favor 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).
In order to detect and understand UHI formation, one must measure the urban climate of a city in a very detailed way, which current official meterological 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 sensing applications to leverage new possibilities of private-owned weather station and sensor networks to interpolate air temperature 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 air temperature between sensors. We show that for interpolating a single sensor, Histogram-based Gradient Boosting is a powerful ML approach that achives a RMSE between 0.4 and 0.5 for sensors from Netatmo and Sensor.Community in Hamburg and Stuttgart, and that other features other than air temperature of surrounding neighbours have little to no influence on the model. Areal interpolation on the other hand seems more challenging and performs much worse with an RMSE between 3 to 4, suggesting that there is more improvement potential in this area.

Table of Contents

List of Figures

List of Tables

1 Introduction

In 2023 56% of the human population already lives in urban areas with the number projected to continiously 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 favor 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 urban heat-island (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 amount 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 Temperature (LST) [PPC+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 microwaces 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 MODIS ranging from $1km^2$ to $5km^2$, while newer satellites such as LANDSAT or Sentinel 2 offer higher spatial resolutions between $10m^2$ to $50m^2$ 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 meterological observation networks, such as operated by the German Weather Service (DWD), offer a much higher temporal resolution at 10 min intervals at 2m and 5cm through the use of 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 center. 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 WHO 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 time 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 networks (PWS) in the case of crowd-sensing meterological data. Well-known examples of professionally setup testbeds include the Birmingham Urban Climate Laboratory (BUCL) [CMY+15] and the Helsinki Testbed [KPS+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 [1] project, or by companies, such as Netatmo [2], 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 quality control (QC) steps [FBD+21, MFG+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, Wundermap, 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 EUMETNET which includes 31 european national meteorological services [HGMS+22].
While these different approaches offer different advantages and disadvantages of measuring air temperature (TA) 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 feasability of the usage of ML models for air temperature interpolation in local urban environments. As part of this exploration, two main use-cases are discussed, namely air temperature interpolation for a single station, and areal interpolation for a wider urban environment. The main idea of air temperature 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 if that sensor is offline, or if there is no sta-

[1] `https://deutschland.maps.sensor.community/`, *last accessed: 22.08.2023*
[2] `https://weathermap.netatmo.com/`, *last accessed: 22.08.2023*

tionary sensor for that specific location to begin with, e.g. a moving 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 air temperature 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 air temperature 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. In order to collect urban weather data, different PWS providers are compared and data collected from Netatmo and Sensor.Community. Data preprocessing steps as well as quality control (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 feasability and identify prediction quality and rank model performances by assessing root mean squared error (RMSE) and r-squared (R2) 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 speare, and interpolation techniques from statistics and geostatistics. In Chapter 3, ML-based interpolation is introduced with a discussion of model selection criteria for air temperature interpolation and a comparison between different ML regression models which can be used for interpolation. Chapter 4 discusses data provider and collection, data preprocessing 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 feasability, model performances, and feature importances. 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 in order to define requirements for the ML models later on. 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, in order to define a baseline for the evaluation of ML-based models.

2.1 Urban Heat Islands (UHI)

UHIs have been the center 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 mayor 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+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+16], mental health [OMPR18], or can disrupt sleep [OMMF17] and causes other issues such as overheating that causes urban infrastrcuture to fail, decreased air quality and low outdoor thermal comfort levels [SJVL+13].

2.1.1 UHI Classification

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

and a vertical 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 planitory/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].

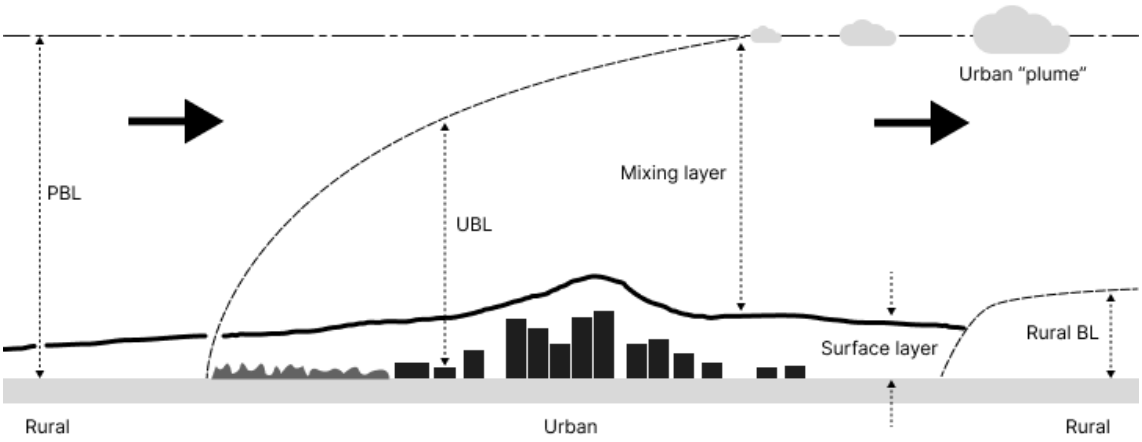

Figure 2.1: Mesoscale view of the urban climate, redrawn from [Oke06]

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 kilometers. It is characterised by the turbulent mixing of air, forming wind currents, that are mayorly influenced by the underlying surface.

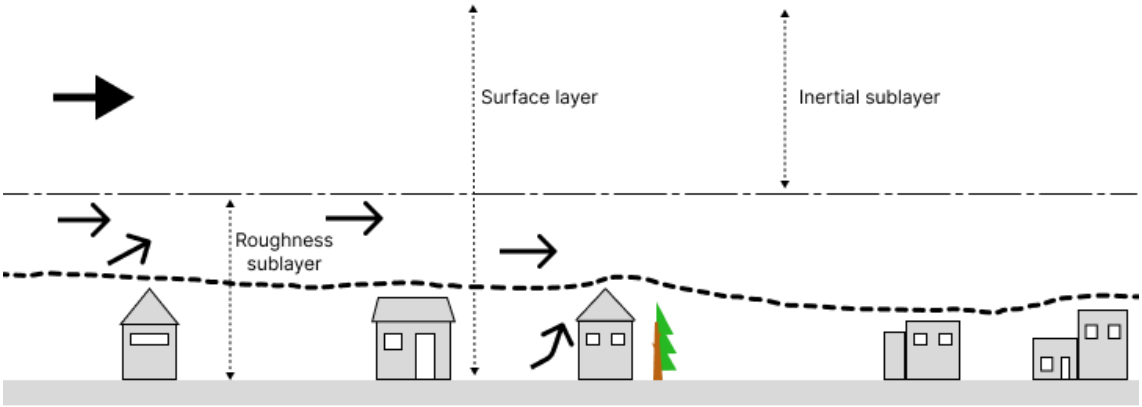

Figure 2.2: Localscale view of the urban climate, redrawn from [Oke06], (Todo finish)

The localscale 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 together to form a more average and representative view of the source area, typically at the scale of one to several kilometers. This

layer is monitored by weather stations that are located at/or sligthly above the canopy height.

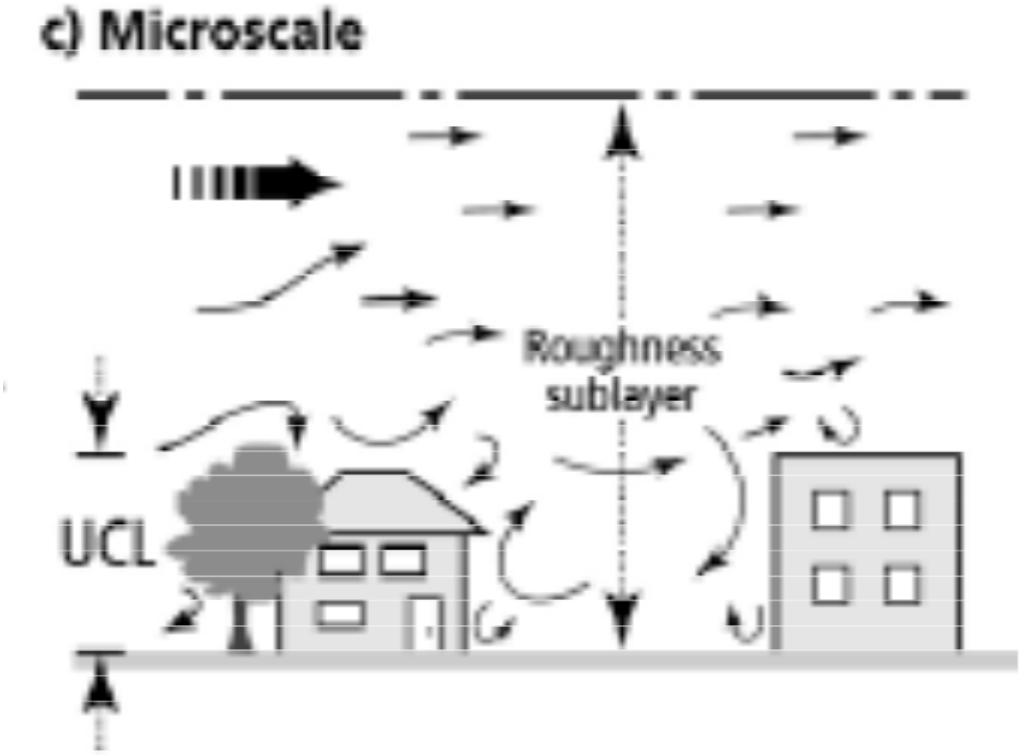

Figure 2.3: Microscale view of the urban climate, redrawn from [Oke06], (Todo)

The microscale usually ranges from a neighborhood scale to individual street canyons or even microclimates created by individual buildings. It is mayorly 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 surface temperature (LST), however the correlation between air and surface temperature varies greatly based on other surrounding influences such as wind velocity or humidity [SB92], especially in cases of extreme temperatures [Goo16]. The closer the air temperature is measured to the surface, the more local the measurement, therefore air temperature of the canopy is usually measured at 2m height, so the different energy fluxes have enough time to mix together and form a more representative average for a bigger area.
Vertically, UHIs can be devided based on these boundary layers into three mayor types [Oke76, OMCV17], namely Boundary Layer 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 differenciating 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+12] focussed on both day and night time UHIs.

2.1.2 Surface Urban Heat Island (SUHI)

The land surface temperature (LST) is measured directly at the surface of an object and is the main indicator of the surface urban heat island (SUHI), which can be found in many cities around the globe [PPC+12]. Surface temperature, in contrast to air temperature, 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 are able to take various measurements. The used sensor has a mayor 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 kilometers, 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 complementay data such as derived vegetation indexes are usually not available as precomputed values, adding additional work to the data retrievel process, as later discussed in Section 4.3. Additionally, weather satellites usually orbit earth to cover wide areas and are not geostionary, 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 emited radiation from clouds [ZPL15], however they also have their limitations. Lastly, LST and air temperature 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 (CUHI)

The canopy UHI 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 air temperature, which is used to measure the urban heat island intensity (UHII) [Oke73], the most commonly used way

of describing the heat island maginitude [KB21].
Since the beginning of modern climatology, mayor 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 shortcommings such as a lack of site desciptions, 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 settings, that require special care due to the huge amount 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+19], therefore such factors need to be taken into account when measuring the UHII.
Compared to LST, TA is measured in situ via weather station networks or other types of sensor networks. Provider for PWS and temperatur 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 densly 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 lead 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 kilometers, 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 encironmental sciences' [SO09]. As a solution, different classification into local climate zones 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 meterological phenomena on the temperatures collected. The urban climate is extremly complex, due to many different influences, such as antropological energy, heat dissapated 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, costlines 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. In order to find out, how a Smart Cities are structured and how applications could take advantage of data provided by a Smart City, we take a look at its general architecture. The most generalised architecture of a smart city consists of four layer, 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+02], an overlay network focused on discoverability while also protecting privacy, with which the data transportation layer could be desinged as a peer-to-peer (P2P) network. Other research focuses on the data accessability 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+19], which could provide crowd-sourced distributed air temperature 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 air temperature, 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.

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

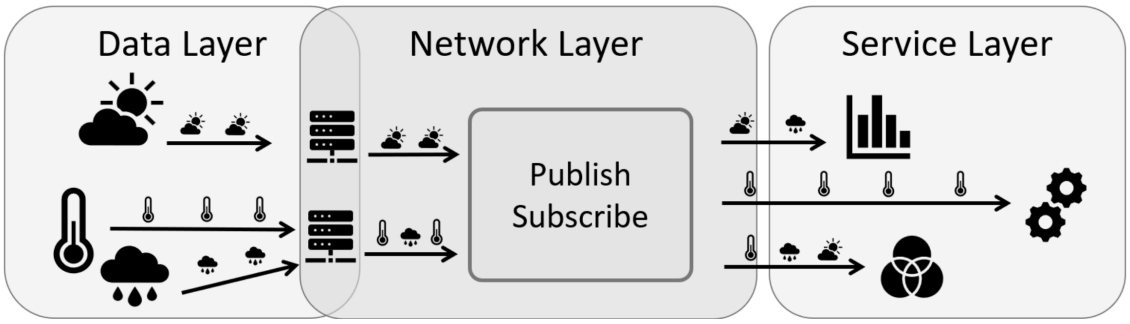

Figure 2.4: In the data layer (left), a wide variety of environmental data is collected with the help of multiple sensors. These are connected to their citizen-owned local base stations, which manage access rights and forward collected data to subscribed services (right) via the decentralized publish-subscribe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

things on the deployment. An ad-hoc WSN has energy and bandwidth contraints due to the usage of batteries as power sources. In contrast, sensors that are permanetly installed, either stationary or on a moving target, and connected to a constant power source don't have this 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 Birgmingham Testbed [CMY+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 a sensor networks by non-professionals [MFG+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+09], although the quality of individual low-cost sensors can be questionable [CDS+17]. Especially PWS data has been used to augment weather data from traditional weather monitoring networks from offical weather services [HGMS+22]. In general, LCSs can improve data availibility 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: air temperature, humidity, atmospheric pressure, reactive gaseous air pollutant (CO, NOx, O_3, SO_2), particulate matter (PM), greenhouse gases (CO_2, CH_4), percipitation, solar radiation, wind speed and direction, anthropogenic heat, noise, sky-view factor, heat fluxes and many more. Correlations between these features can vary greatly based on surrounding factors. In order to better understand these correlations, many em-

pirical studies have studied the influence of meteological 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 air temperature, the sampling rate could be an average temperature measured over five minutes, whereas percipitation 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 callibrated reference-grade high cost sensors and might have lower response rates, and perform better or worse based on the meteological 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 generelly 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 representativness 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. On 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 air temperature sensing, LCS could be mounted to moving vehicles such as busses, cars, e-bikes and 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 surounding sensors so that air temperature could be interpolated for times when a moving sensor is currently not a 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 sen-

sors are more robust in measuring short extreme events such as heat waves.
Next to mobile sensor networks, offical weather services such as the DWD also do temperature profile measurements [1], where sensors are mounted to a car to capture air temperature, 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.' [2].

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 build 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 otherwiese 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 generelly 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 air temperature 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 air temperature. 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 (crowdsourced) 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.

[1] `https://www.dwd.de/DE/service/lexikon/Functions/glossar.html?lv2=101996&lv3=102106`, *last accessed: 23.08.2023*
[2] `https://www.dwd.de/DE/forschung/klima_umwelt/klimawirk/stadtpl/projekt_warmeinseln/projekt_waermeinseln_node.html`, *last accessed: 23.08.2023*

2.3 Interpolation

Interpolation is in essence to determine unknonwn 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 and 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 really specifc 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 volume-preservance.
Generally, spatial interpolation methods can be categorised by many different factors. Lam differenciated 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-situi 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 mayor 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 it's simplicity and fast computation, inverse-distance weighting (IDW) is a popular and commonly used interpolation method. The main downside with IWD is that it is a smoothing technique, therefore it is not able to capture local maximas and minimas, 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 caluclate 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 itteratively.
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 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 Krigin methods available, each suited for different use cases, as offered by ArcGIS [3] including:

1. Simple Krigin: the simplest form of Krigin, that assumes that the mean of the measured values is known and constant
2. Ordinary Krigin: same as Simple Krigin, but the mean is an unknown constant
3. Universal Krigin: instead of assuming a constant mean, the mean is modeled as a deterministic function

[3] `https://desktop.arcgis.com/en/arcmap/latest/extensions/geostatistical-analyst/what-are-the-different-kriging-models-.htm`, *last accessed: 24.08.2023*

4. Indicator Krigin: same as Ordinary Krigin, but for categorical data
5. Propability Krigin: same as Indicator Krigin, but assumes two types of random errors that can are each auto-correlated and cross-correlated to each other
6. Disjunctive Krigin: same as Ordinary Krigin, 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. Cokrigin: offers methods for the previous Krigin 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

In the context of geostatistical analysis, there are different types of Krigin methods available that combine the aforementioned methods with other techniques, such as regression analysis. The following list is the geostaticial methods offered by ArcGIS Pro as part of the Geostatical Analyst toolbox [4]:

1. Empirical Bayesian Krigin (EBK)
2. Empircal Bayesian Krigin 3D (EBK3D)
3. EBK Regression Prediction (EBKRP): Empirical Bayesian Kriging with regression prediction
4. Global Polynominal Interpolation
5. Kernel Interpolation with Barriers
6. Moving Window Krigin
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 *PyKrige* [MYM22], which are the following:

- Ordinary Kriging
- Universal Kriging
- Regression Kriging

[4] `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 AI, transformer-based neural networks have revolutionised text and image generation. Models such as OpenAI's *ChatGPT* [Ope23] or Google's *LaMDA* [TDFH+22] have generated significant hype for the possiblity 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 continous 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 air temperature 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 air temperature interpolation.

AI vs. Machine Learning vs. Deep Learning

Before diving deeper into the applications of ML, we need to clarify what is meant by artificial intelligence (AI), machine learning (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 networks (ANN), 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 Temperarture Interpolation

As meteorological research and analysis activities are usually in need of gridded or continuous data [SKP+20], interpolation is a really important tool to convert single data points at distinct locations from ground-based weather stations into a continous layer. Interpolation can also be applied to individual sensors in order 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 air temperatute 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 time-frames 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 air temperature maps, a sensor density of at least 1 sensor per km^2 is needed [VBEM20], however the denser the sensor network the better the prediction quality, as even inside a single street canyon air temperatures can easily vary by 2 to 3°C [SHN+08]. In order to achive 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+22].
In the context of this work, we discuss both ML applications and try to show the feasability and potentials of ML-based interpolation in the context of urban air temperature sensing. In the following, we introduce several ML algorithms that can be used for regression tasks and discuss how they need to be adapted in order 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 in order 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 veariable 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 percipitation influences the humidity. Other assumptions could be the distribution of data, the mean of values not changes, etc.
- **Accuracy and Reliability:** Creating an accurate and reliable model is really 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 choosen 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 f.e. 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 f.e. 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, Neural Networks (NNs) 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 additonal data preprocessing 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:** Similar to missing data, handling of sparse data is also really 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 available at all times. 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, generally speaking, the more training data it needs to fine-tune all it's weights and the longer it takes to train, either due to the amount of data or the amount 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 the handle massive amounts of data, live retraining or sophisticated support via ML libraries such as *Tensor Flow* [1] 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 reinforment, or the type of problem they try to solve, e.g. classification, regression or clustering. The most important differenciation 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 (continous) target variable y based on a set of input variables X, like it is the case for temperature interpolation. Generally speaking, this problem can be classified as a supervised learning problem, therefore possible algorithm candiates contain the following models:

- (Multi) Linear Regression (MLR)
- K-Nearest Neighbours (KNN) Regression
- Regression Trees and Forests
- Histogram-Based Gradient Boosting
- Support Vector Regression
- (Deep) Neural Networks
  - Long-Short Term Memory (LSTM)
  - RNN

[1] https://www.tensorflow.org/

Next to these algorithms, there also exist less popular regression algorithms, 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+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 air temperature 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 air temperature interpolation, regression forests and histogram-based gradient boosting seem to be popular choices and perform better compared to other methods [AYDK22, HKS+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 continous dependent variable based on a number of independent variables. These independent variables can be either continous or discrete. The model can be expressed as follows:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \epsilon \tag{3.1}$$

where y is the dependent variable, x_1 to x_n are the independent variables, β_0 to β_n are the parameters of the model and ϵ is the error term. The relationship between the parameters is assumed to be liniear, 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 4.5.
Linear regression has the advantage, that it is a very simple model, therefore the majority of 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. In order to fit no-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 with 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 in order to find the best trade of between bias and variance. The model can be expressed as follows:

$$\hat{y} = \frac{1}{k} \sum_{i=1}^{k} y_i \tag{3.2}$$

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.
KNN is part of the family of non-parametric models, meaining 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. Due to the fact that 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 combine multiple tree predictors and train them on different features and subsets of data and either average their predictions in order 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 continously 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 continous target variable such as mean squared error (MSE). The model can be expressed as follows:

$$\hat{y} = \sum_{m=1}^{M} c_m \mathbb{1}(x \in R_m) \tag{3.3}$$

Regression trees are comparitively 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 air temperature 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 Neural Network and Deep Learning 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 R2 values achived by related studies.
Ho et al. [HKS+14] mapped maximum daily air temperatures for Vancouver, Canada on hot summer days and combined remote sensing TM/ETM data from Landsat with field observations from Environment Canada and Weather Undergorund (WoW). They compared Ordinary least squares regression, SVM regression, and Random Forest Regression and achived the following RMSE:

- Random Forest: 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+18] RFsp [HNW+18] Random Forest for spatial prediction Random Forest for spatial Interpolation (RFSI) [SKH+20]

There are a couple of studies that use regression trees for temperature interpolation. In [VBEM20] the authors achive an average RMSE of 0.52 °C (R2 = 0.5), 1.85 °C (R2 = 0.05) and 1.46 °C (R2 = 0.33) for annual mean, daily maximum and minimum air temperature respectively for the city of Oslo, Norway by combining 20 features from satellite data and PWS from the Netatmo network.

In [ZKBK21] the authors used the quantile regression forest algorithm use regression trees to predict the daily maximum and minimum air temperature for the city of Zurich, Switzerland. They achieve an average RMSE of 1.5 °C (R2 = 0.7) and 1.4 °C (R2 = 0.7) for the daily maximum and minimum air temperature respectively by combining 20 features from satellite data and PWS from the Netatmo network.

There is a systematic bias to underestimate high temperatures and overestimate low temperatures [ZKBK21, ZL12]

3.3.4 Histogram-Based Gradient Boosting

Similar to Random Forests, Histogram-Based Gradient Boosting (HGB) 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 Random Forests, HGB as implemented by sklearn [2] based on LightGBM [KMF+17], should

[2] `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingRegressor.html`, *last accessed: 25.08.2023*

have better performance, especially for bigger datasets and higher dimensional features, has build-in support for missing values which simplifies data pre-processing steps, and finally seems to slightly outperform RF according to the sklearn documentation [3] and other studies [AYDK22].

Bagging Methods

In order 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]

Due to the popularity of RF as an extension of bagging methods, we do not consider these bagging methods further in this work.

3.3.5 Support Vector Machine Regression

Support Vector Machines (SVM) 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 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 most simple ANN is the perceptron [Ros57] which models one single neuron, consisting of one or many inputs, a single processor,

[3] https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_hist_grad_boosting_comparison.html

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. Generally speaking, 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 (DAG) 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 the ML method in this work, however want to keep in mind that ANNs could be a powerful tool for tasks such as LST and air temperature estimation and prediction [YSL+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 take a look at potential features for air temperature, 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 (NPL) and computer vision (CV), 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 [1]. 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 distrubuted or often not openly available. Platforms such as Google Earth Engine [GHD+17] try to address this issue, however the dataset catalog [2] 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 f.e. 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 refrence 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+13] such as the Helsinki Testbed [KPS+11] or the Birmingham Urban Climate Laboratory [WYC+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 often

[1] `https://research.google/resources/datasets/`, *last accessed: 05.08.2023*

[2] `https://developers.google.com/earth-engine/datasets`, *last accessed: 05.08.2023*

only run for a limited time 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+09]. In the context of meterological data, such such sensor networks are often referred to as citizen weather station (CWS) [MFG+17] or personal weather station (PWS) [HGMS+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 air temperature interpolation. In the following sections, we take a look at available PWS providers and their data, look at potential features for air temperature interpolation, discuss additonal 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 such as Netatmo, or provide guides to allow users to connect their own sensors to the platform such as Sensor.Community. Netatmo data in particular has been used in several studies [MFG+17, HGMS+22, VBEM20, ZKBK21] and has seen complementary studies for example discussing QC
processes [FBD+21], later seen in Section 4.4. There are also other PWS network providers such as Sensor.Community or Weather Underground (WOW) that have been used in several studies [HKS+14]. In order 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 [3] is a contributers driven global sensor community that creates Open Environmental Data, and has an archive [4] of their historical sensor data world wide. There are no quality measures recorded for each sensor, but as crowd-sourced sensor data

[3] `https://sensor.community/en/`, *last accessed: 05.08.2023*
[4] `https://archive.sensor.community/`, *last accessed: 05.08.2023*

tends to have a lower quality than professsionally setup sensors, e.g. sensor placement by non-professionals, we need to explore how the data quality looks like. In Figure 4.1, 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 adressed in the data pre-processing step and can result in a significant reduction of available data. This was also an issue dicussed in [MFG+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.

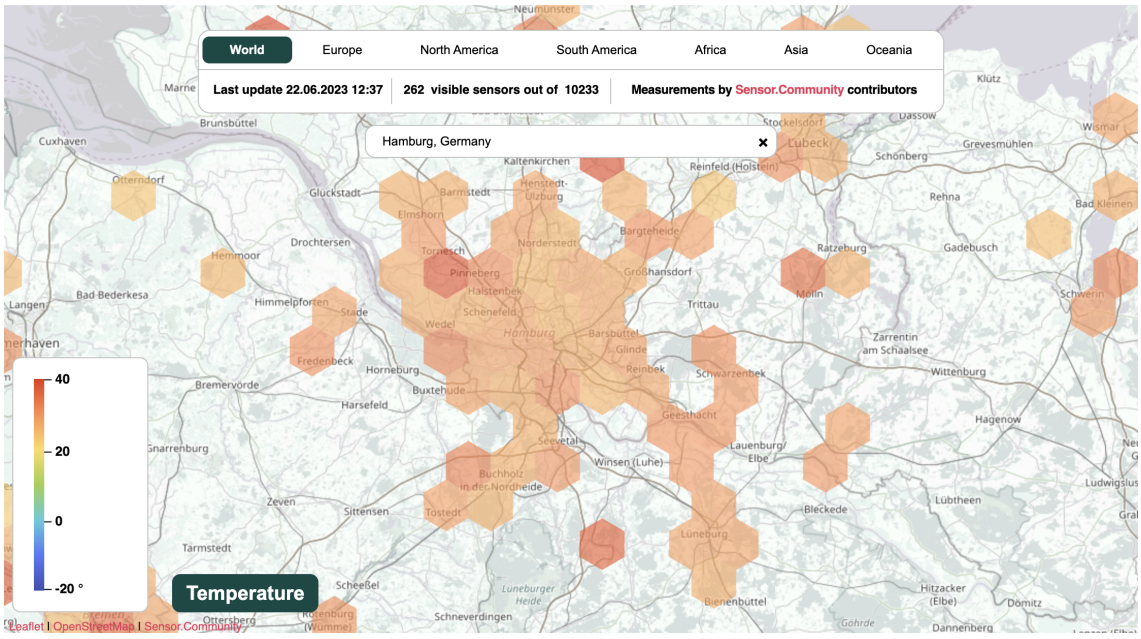

Figure 4.1: Temperature map from Sensor Community for Hamburg, Germany, on 22.06.2023 12:51h with the DWD reference at 25°C

Sensor	Temperature °C
Median 2 Sensors	37
#32708	50
#64198	25

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 [5]. Of these sensors, many are located in Germany, as seen in Appendix 3, and almost half of them are of type BME 280, which is a lost-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.

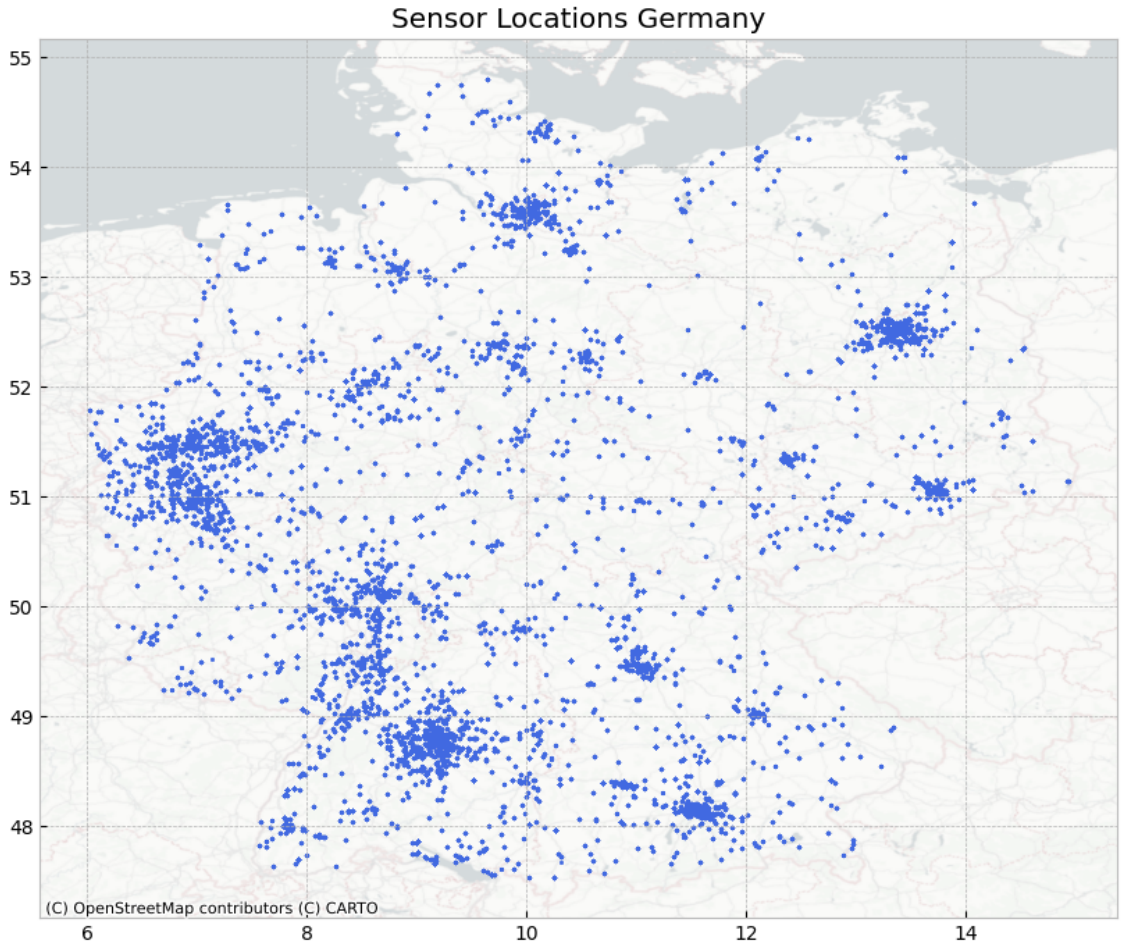

Figure 4.3: Sensor locations of Sensor Community in Germany, as of 01.05.2023, of sensor type DHT22 (2590 sensors), BME280 (1558 sensors), BMP280 (100 sensors), BMP180 (72 sensors)

4.1.2 Netatmo

Netatmo [6] 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 [7] where customers can share their outdoor weather station

[5] as of 24.06.2023
[6] `https://www.netatmo.com/en-eu`
[7] `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)

data. They provide an API to access current weather station data as well historic data from individual outddor 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 [8] which is a network of 31 European meteorological and hydrological services (NMHSs). 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+22]. There are currently no openly historical datasets available from Netatmo data, only private datasets [9] that are only available for partners such as EUMETNET members. They offer an educational program [10] to acess temporally and spatially limited amounts of data that is usually only available to commercial partners.

In the context of collecting meteological data, the smart weather products are of particular interest. These include a smart outdoor weather station that collects air temperature, 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 a large amount of 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 [11] 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 hours and 50 requests every 10 seconds per user [12]. In this work, we use the REST API to collect sensor data from Netatmo sensors.

[8] `https://www.eumetnet.eu/`, *last accessed: 06.08.2023*
[9] `https://catalogue.ceda.ac.uk/uuid/e8793d74a651426692faa100e3b2acd3`, *last accessed: 06.08.2023*
[10] `https://www.netatmo.com/en-eu/weather-with-netatmo`
[11] `https://dev.netatmo.com/apidocumentation`
[12] `https://dev.netatmo.com/guideline#rate-limits`, *last accessed: 06.08.2023*

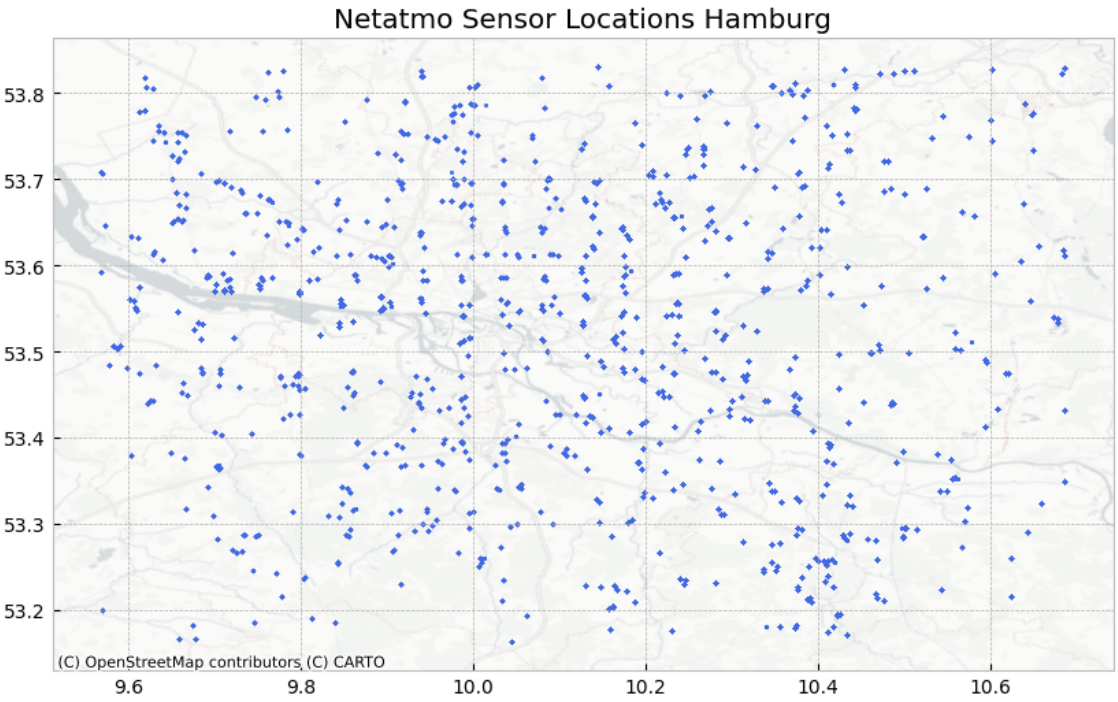

Figure 4.4: Sensor locations of Netatmo in Hamburg, Germany, as of 28.06.2023

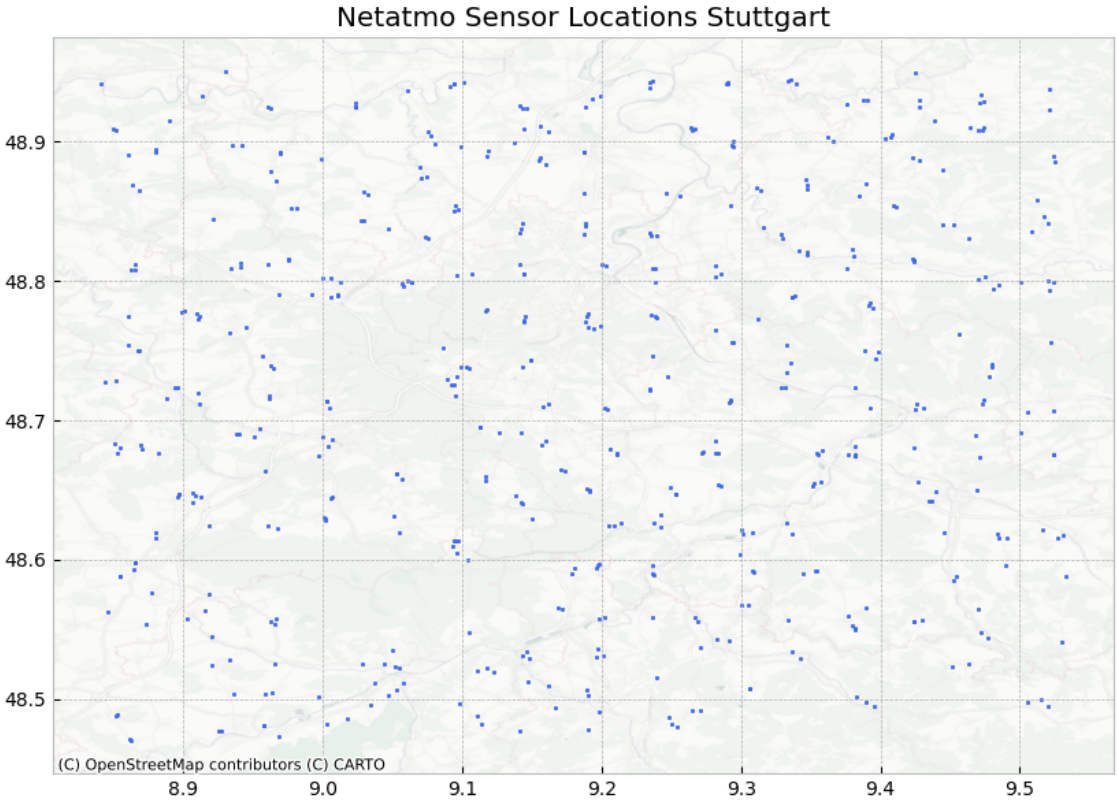

Figure 4.5: Sensor locations of Netatmo in Stuttgart, Germany, as of 19.06.2023

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 resultiing in a proneness to radiative errors, and therefore overall slow sensor-response time [MFG+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 build 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 WeatherObservationsWebsite (WOW) [13] and Weather Underground [14]. 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 for this work.

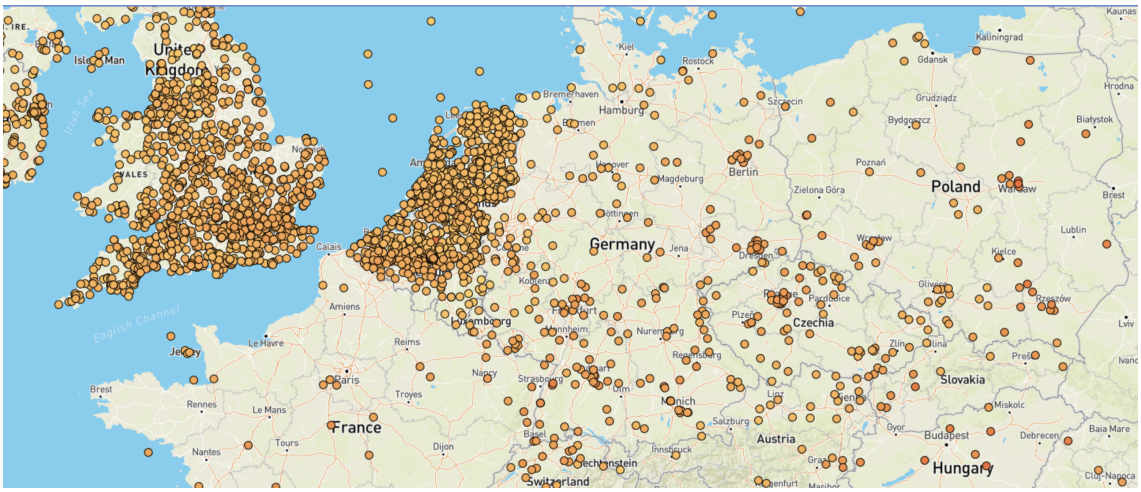

Figure 4.6: Temperature sensor locations from WOW, acessed on 05.07.2023

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

[13] https://wow.metoffice.gov.uk/
[14] https://www.wunderground.com/

to current World Meterological Organization (WMO) guidelines [WMO18] in order to ensure high data quality. These standards are either achived by offical weather services, or by institutions such as universities whose sensors are maintained by experts.

4.2.1 DWD

The official german weather service (DWD) has many objectives, that are defined by the DWD-law in Germany. Its tasks include meterological and climatological monitoring of the atmosphere, meterologically securing the airspace for civil aviation, monitoring the maritim climate, and more. The DWD operates a large monitoring network and publishes most of it's data via its OpenData portal [15].
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 UHI intensity [16]. 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 [17] that are used to measure precipitation and wind speed. This data could be interesting in the context of air interpolation

[15] `https://opendata.dwd.de/`, last accessed 13.07.2023
[16] `https://www.dwd.de/EN/climate_environment/climateresearch/climate_impact/urbanism/urban_heat_island/urbanheatisland_node.html`, last accessed 12.07.2023
[17] `https://www.dwd.de/DE/leistungen/radarprodukte/radarprodukte.html`, *last accessed 12.07.2023, not available in english*

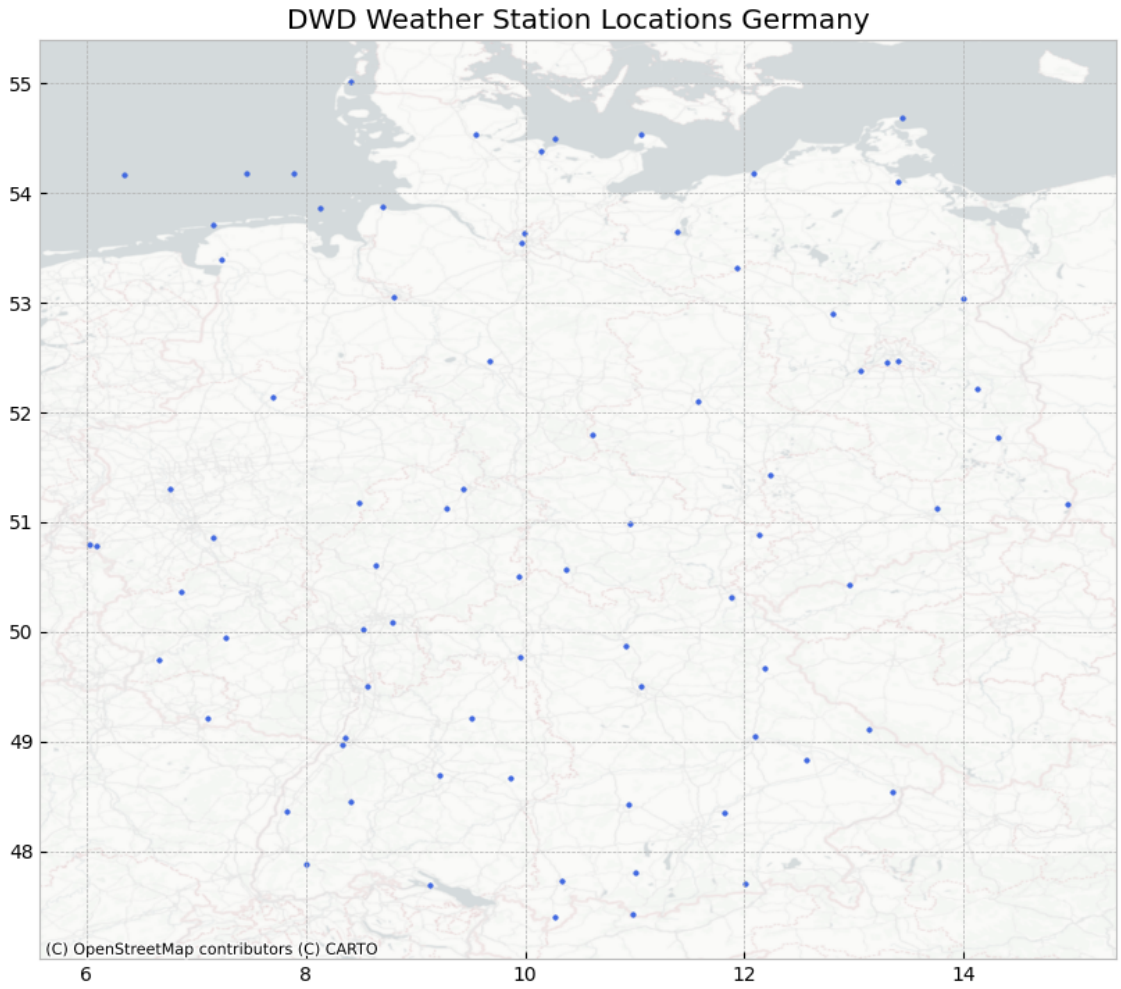

Figure 4.7: DWD Weather Station Locations in Germany, `https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/subdaily/standard_format/KL_Standardformate_Beschreibung_Stationen.txt`, accessed 28.06.2023

in order to detect precipitation events that have a mayor influence on humidity and temperature or to detect wind speeds that also play an important factor in dissapating heat 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 offical weather services, many public and private institutions operate weather stations. The following section lists two examples of such institutions, the Meterological Institute of the University of Hamburg and the Office for Environmental Protection of the City of Stuttgart.

University of Hamburg – Meterological Institute

- address local climate concerns and support decision making - Projekt HUSCO (Hamburg Urban Soil Climate Observatory)

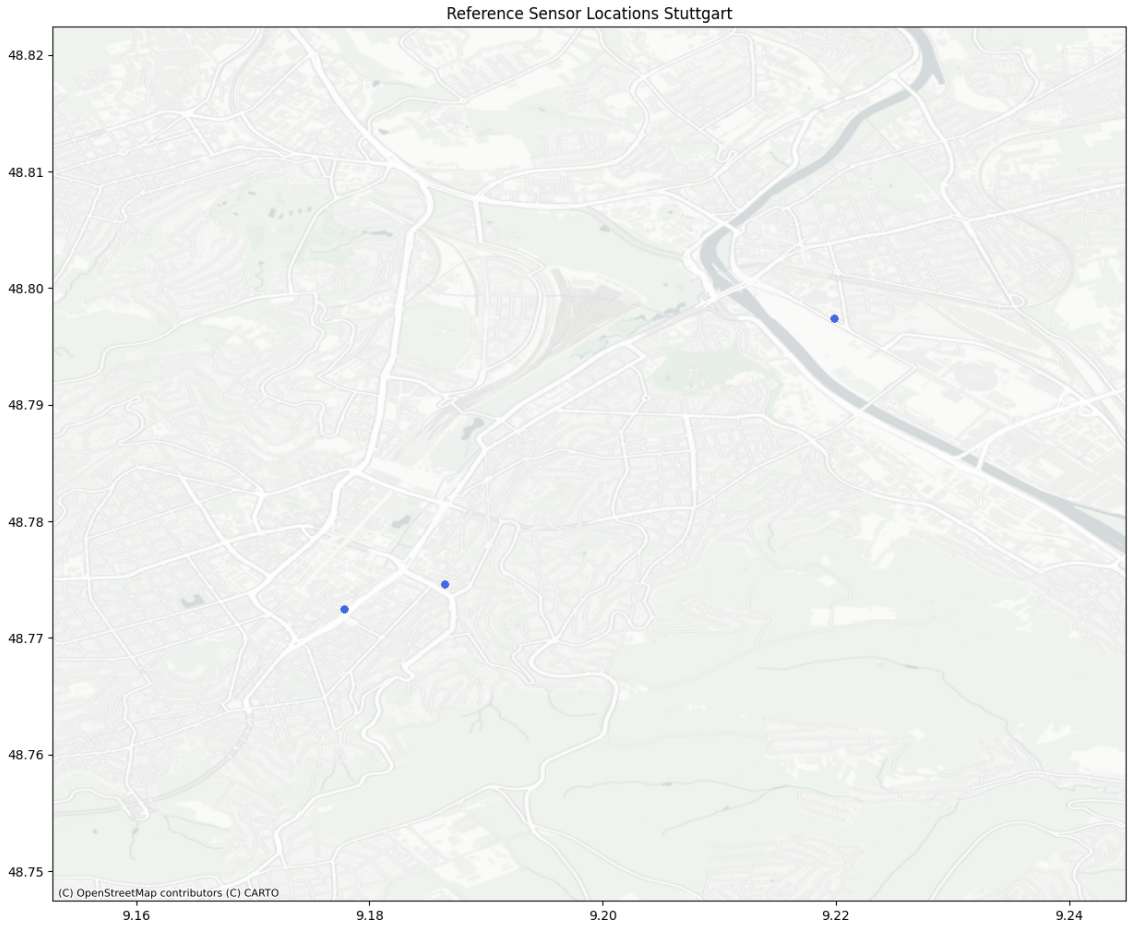

Figure 4.8: Weather Station Locations in Stuttgart, `https://www.stadtklima-stuttgart.de/`, *last accessed: 10.08.2023*

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 air temperature 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 air temperature 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 meteological data via satellites, planes or ballons 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 surface temperature, e.g. LST data. 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 anaylsis of weather phenomena. Another upside is the great data availibility, as many satellite providers, such as ESA..., publish their satellite data. This creates many research opportunities, and many services directly rely on these measurements (todo list).
Remote sensing also comes with some 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. As a consequence, satellites only pass over each individual area a couple times a day, making real-time applications for currently unobserved areas impossible. Additinally, the spatial resolution can be too low for micro-/local-scale analysis, with typical LST resolution spanning from 1 km^2 to tens or even hundreds of 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, traditionally making measuring under cloudy/rainy conditions either impossible (example) or less accurate (example), by instead relying on outgoing radiation from the surface. These restrictions highly depend on the sensor used, as different sensors use different sensors use different technologies, e.g. microwaves with different wave lengths or higher resolution sensors.

4.3.1 Google Earth Engine

Todo: Explain usage

4.4 Quality Control

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 air temperature 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 erronous 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 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+22]. The following tools are available:

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

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

[18] https://doi.org/10.14279/depositonce-6740.3
[19] https://github.com/dafenner/CrowdQCplus
[20] https://github.com/metno/TITAN
[21] https://source.coderefinery.org/iOBS/wp2/task-2-3/netatmoqc

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+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 heigh check due to the missing values. Important to note here, that in the following, only the air temperature is validated and not 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 air temperature 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 air temperature. 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, f.e. 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 accross Germany, therefore comparing smaller areas could be beneficial for this. We can also see, that the m5 check, which 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. For the air temperature, 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

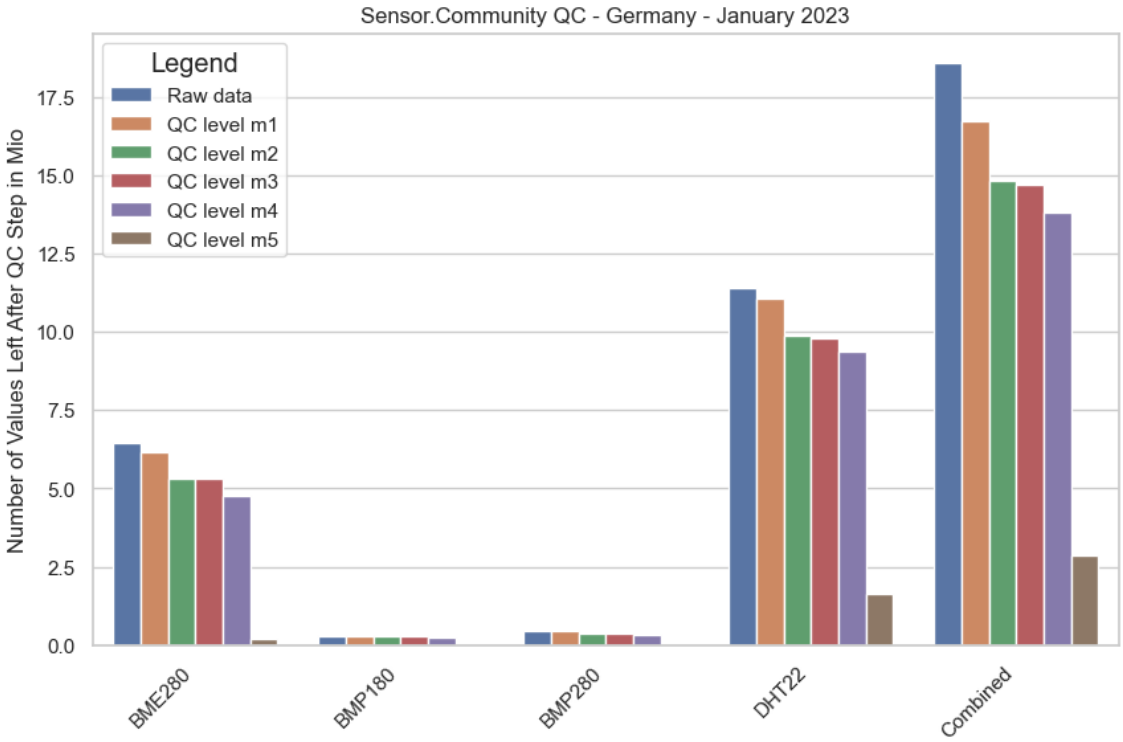

Figure 4.9: QC Results for Sensor.Community Data for Germany, January 2023

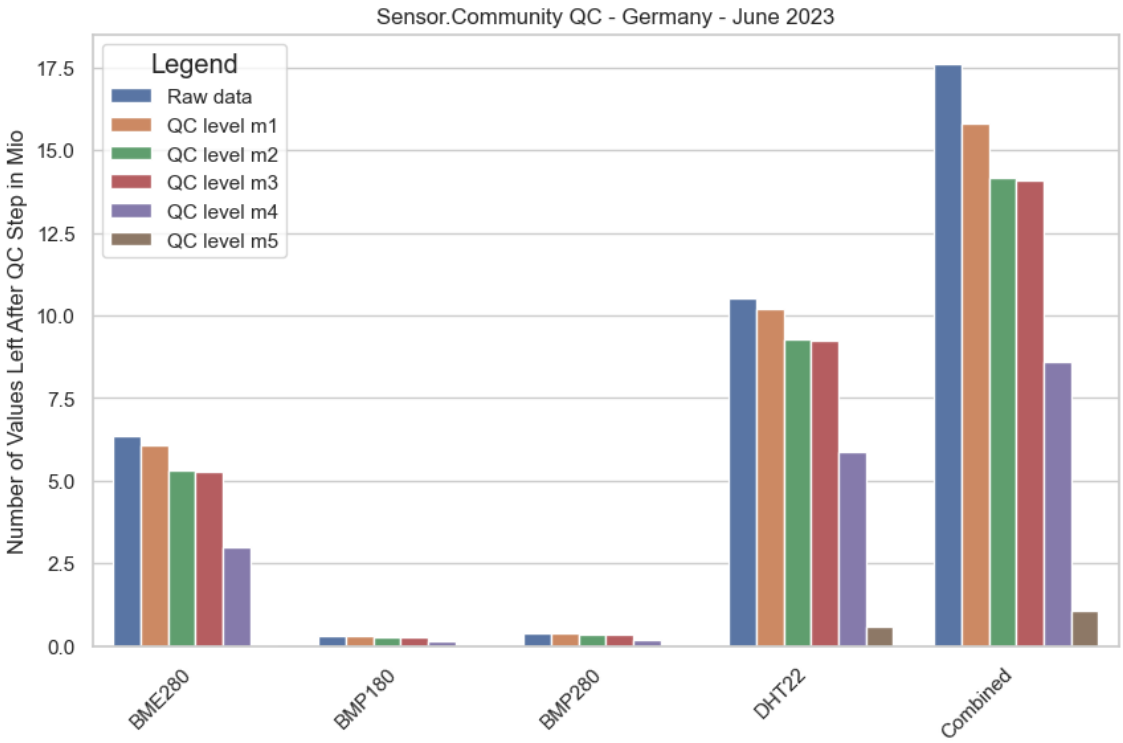

Figure 4.10: QC Results for Sensor.Community Data for Germany, June 2023

sophisticated QC process should be used. In additon, due to the low sensor density and the fact, that all types of sensors used are good low cost sensors, 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.

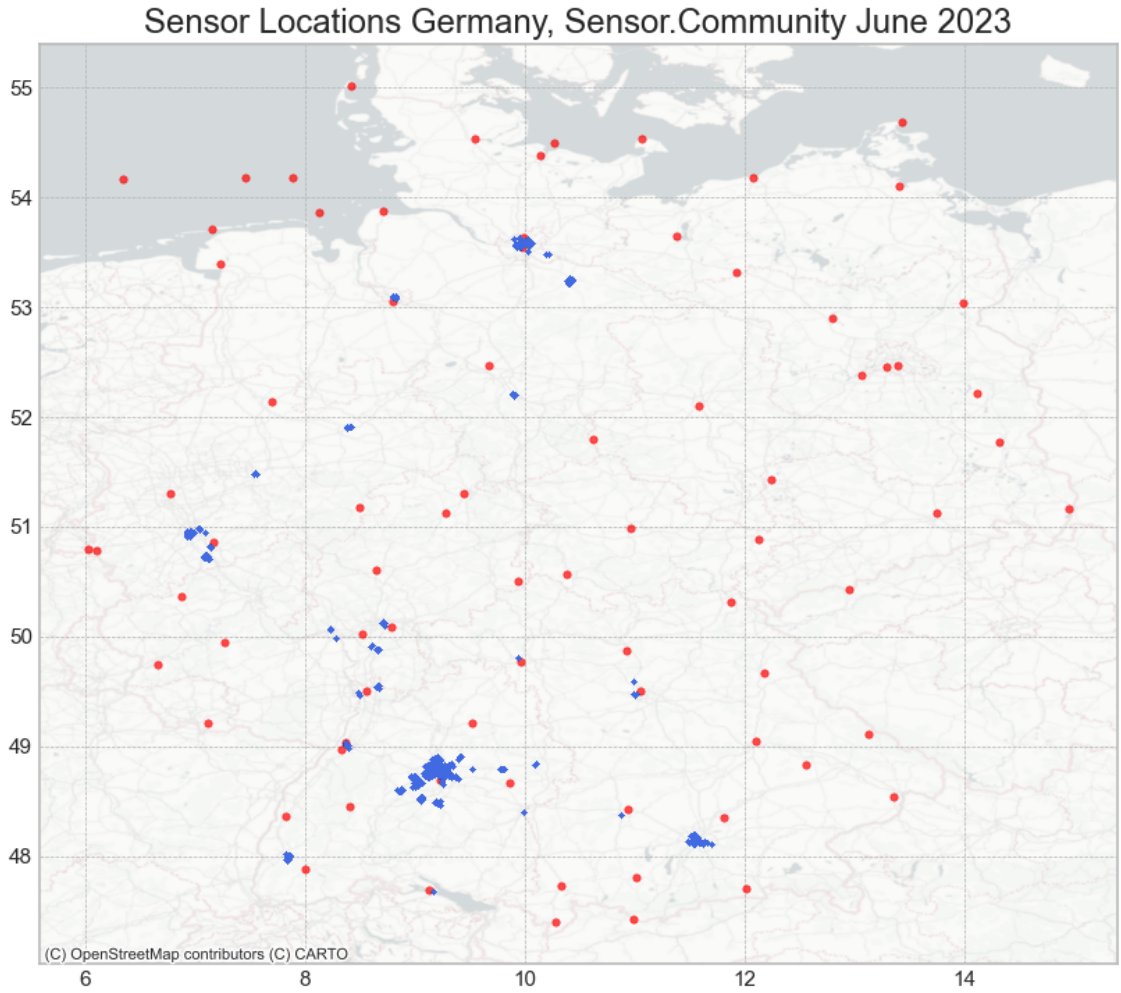

Figure 4.11: QC Results for Sensor.Community for Germany, June 2023

4.4.2 Quality Control for Netatmo

The QC results for Netatmo stations in Hamburg during June 2023 can be found in 4.12 and 4.13. 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+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

Table 4.2: Quality Control Steps of CrowdQC+

	Id	Name of Step	Functionality	% of Data	Num Stations	Num Values
Required						
	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	-	-	-
	o4	Correction for Time Constant	Sensors have different times that they respond to ta changes. 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.

with CrowdQC [MFG+17]. It is interesting to note that stations directly next to water seem to be removed proportinally more often. This is could be due to higher variability of wind close to water [HKS+14] which can result in higher prediction errors.

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 compared to 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. Especially in the context of air temperature 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 air temperature at canopy height, e.g. 2m height. The input features are a combination of sensor readings from weather stations, sensor networks such as Sensor Community and Netatmo, satellite data such as land cover and vegetation health or LST, and additonal meta data such as soil conditions or zoning plans. The goal of this section is to give an overview of the different features that can be used for air temperature interpolation and discuss several highly important features that are especially relevant in the context of urban microclimate.

Next to the model, the most important thing for a machine learning algorithm is the data. Even when the model is perfectly suited for the task at hand, if the data is not suitable, e.g. not enough data, wrong quality, wrongly prepared or formatted etc., the model will not be able to perform well. In the following, we take a look at the data coming from the data-layer and discuss important assumptions such as spatial and temporal autocorrelation. These correlations are important to consider, as they could invalidate models as f.e. Linear Regression 3.3.1 assumes uncorrelated input variables.
First of all, the data-layer exposes a variety of single data-points for various features for different locations for current and past points in time. .Features other than air temperature could further improve the predecition quality, like how [AR20] suggests that in their study the Normalized Difference Vegetation Index (NDVI) and Modified Normalized Difference Water Index (MNDWI) have a strong impact on their estimation model. Therefore, we discuss how additional features can be included in the model.
This model should then be deloyed inside the *service-layer* and act as a building block for further temperature related research and analysis, as air temperature is an important variable for research in agronomy, meteorology, hydrology, ecology and many other fields of application and could be used for UHI detection in the context of smart cities. The general idea and architecture behind the model should not only be applicable for air temperature, but also other types of output features, even tho potentially significant domain knowledge, like in geostatistical analysis and statistics, is required to select and

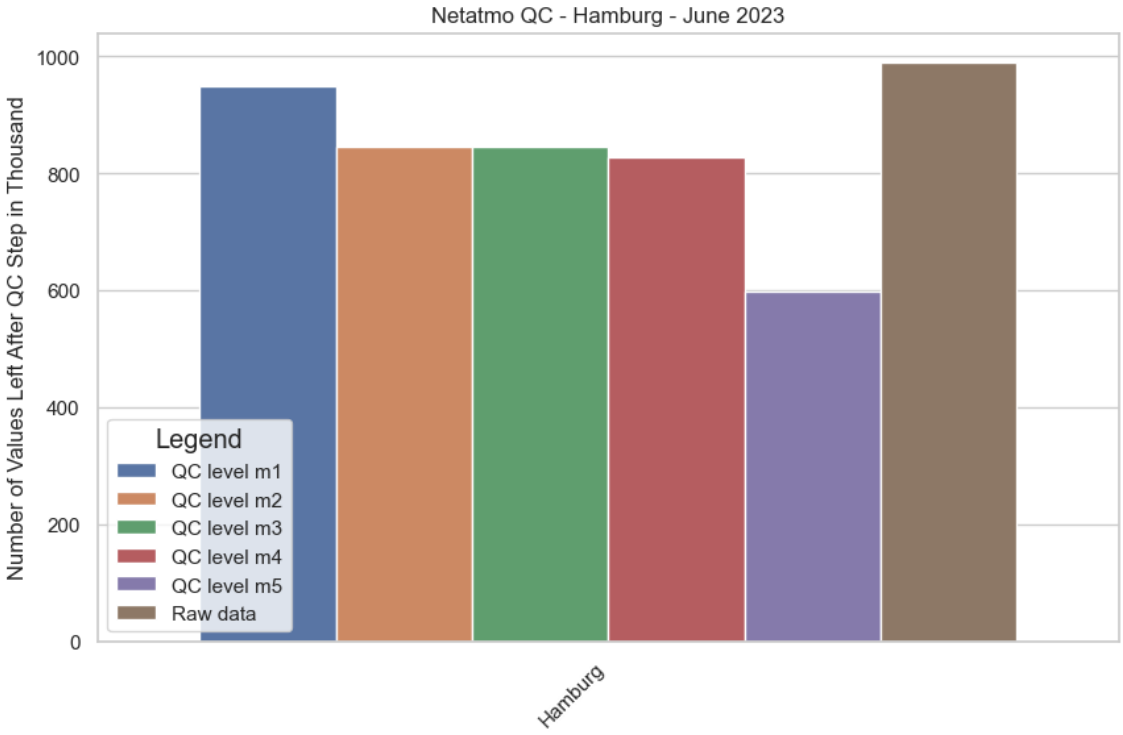

Figure 4.12: QC Result Statistics for Netatmo Data for Hamburg, June 2023

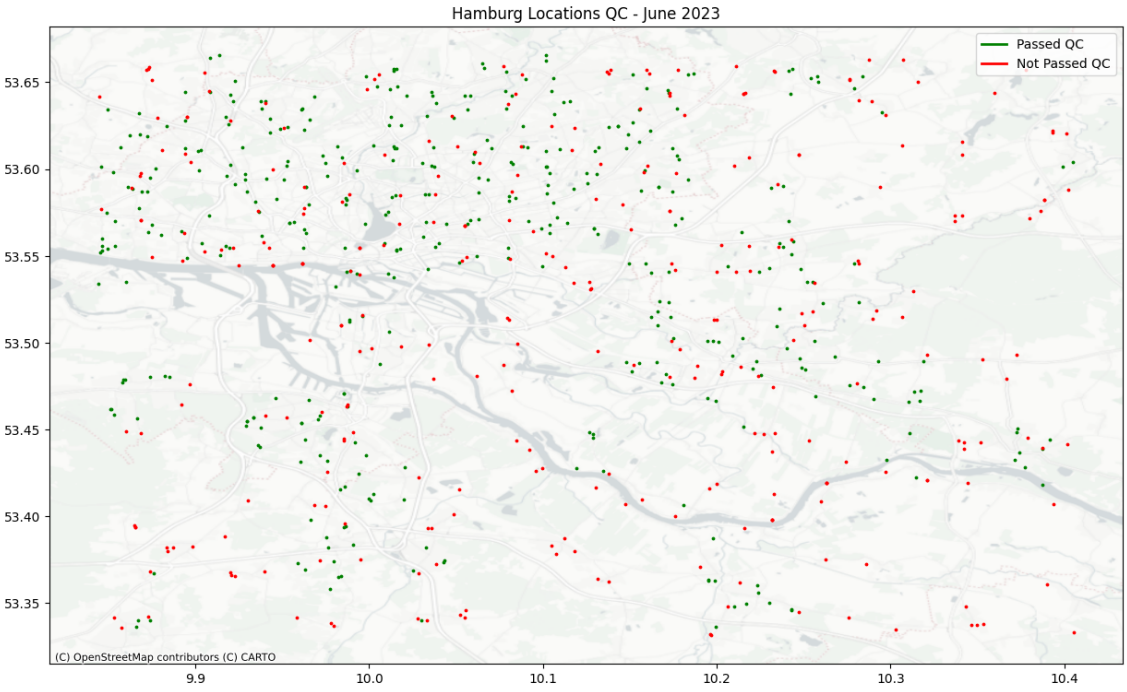

Figure 4.13: Netatmo Stations for Hamburg, June 2023

prepare the input features.
Generally, there are different types of input features. Firstly, there are discrete measurements that capture the underlying continuous geological process as a specific value at a certain point in time. Examples for this are the measurements from sensors which might be deployed in an urban environment to capture air temperature, humidity and more. Important to note here is the interval, the values are captured. For example, a sensor might capture the air temperature as the average over five minutes, whereas a rain sensor might capture the absolute amount of rain in a specific time interval. Secondly, there are calculated features, which might be derived from the discrete measurements or the location of a measurement. For example, the distance to the closest body of water with a minimum size of x^2 meters might be a important information for the air temperature. Lastly, ...
In the following chapter, we discuss how different types of correlation between measurements can be taken into account to improve the model and gain additional insights into local (meteological) dynamics.

Spatial Autocorrelation

As discussed in chapter 2, there is a dependency between air temperature and other meteological features and the location of the sensors. The closer a sensor is to another sensor, the more correlated the sensor readings should be. In geo-statistics, this is called spatial autocorrelation and can be defined traditionally with the Moran's I index [Mor48] or the Geary's coefficient [Gea54].

This relationship is however greatly influenced by the type of feature and the location of a sensor. As explained in [Oke06], the sensor placement in the urban environment is a complex task, as the urban environment is highly dynamic and sensors can be influenced by highly local phenomena, such as air temperarture by heat vents or solar radiation by buildings and surface materials. For this model, we assume that the sensors are placed in a way that the sensor readings are representative for the area they are placed in, even tho in practice this assumption might not hold up, especially for sensors placed by non-experts.
Each sensor reading from the data-layer has a location associated with it in the form of longitute, latitude and altitude (if available). This information can be used to calculate the distance between sensors and subsequently the correlation between sensor readings. However, how exactly the distance between two locations is calculated is not trivial. As the earth is a sphere, the distance between two points on the surface is not a straight line. Depending on the application, different distance metrics can be used. For a small area, such as the city of Hamburg, the euclidean distance could be sufficient, as the curvature of the earth for a small distance is negligible. However, for larger areas the geodesic (harvesine) distance might be more appropriate. In this work, we are focussing on a single city including it's surrounding area, therefore the euclidean distance should have a

sufficient accuracy while also simplifing the calculation. As the city of Hamburg does not have big differences in elevation, the altitude is not considered in the distance calculation.
The location data can be incorporated into the input data in several ways. The most straight forward way would be to just include the longitude and latitude as input features. Depending on the type of model used, these values might need to be normalised. For example, tree-based models do not require normalisation (cite), as they are not sensitive to the scale of the input features. However, neural networks are very sensitive to the scale of the input features and therefore require normalisation (cite). This approach has the downside, that the distance between sensors is not directly encoded in the input data. If this information is important for the model, it could be included by precalculating the distances between all sensors, however this would increase the complexity of the model especially for large amounts of sensors, as this would result in a quadratic number of input features.
The next approach would be to convert the

- area n x m grid of data points, each grid cell with n x n size (depending on the resulution), 4D space with location as x and y coordinates (euclidean distance -> but only for smaller areas such as city of Hamburg and suroundings), time as z coordinate and feature values as w coordinate.

Temporal Autocorrelation

- trends -> expect normally distributed temperature curve -> temporal lag (e.g. temperature of yesterday has influence on temperature of today)

Either explicitly model

Temporal Cross Correlation

Dealing with Correlation

Todo: Techniques to turn correlated variables into uncorrelated ones - principal component analysis (PCA) - ...

variance inflation factor (VIF)
-> over 10 = invalid model[MPV21] -> others more moderate with max 3 [ZIE10]

Dealing with Uncertainty

- The model has a lot of uncertainty - model uncertainty in input data? (depending on sensor type, sensor age, placement...)

- dealing with bias - dealing with variance -> problem of over-fitting

4.5.1 Feature Overview

Essential Climate Variables

Essential Climate Variables (ECV) are a list of currently 50 variables that are proposed by the WMO to measure climate and climate change. The WMO regularily 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 air temperature and land cover. There are three categories of ECVs, namely Atmosphere, Land, and Ocean. And overview of the ECVs can be found online [22].

Hamburg ICDC overview: - atmospheric data - air temperature - pressure - wind - reciptation - clouds - aerosols - humidity - radiation - climate indices - ocean - water temperature - wave height (SSH) - salt content - tide - ocean color (e.g. plankton etc.) - 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 - perma-frost - society - social science parameters

Features used in Related Work

Related studies can give a good overview of which features work best for air temperature interpolation. The used features can be roughly divided into three categories: in-situ measurements, satellite data, 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 4.3 to predict air temperature.

These indexes are either available as precalculated datasets, or can be calculated from raw satellite data. Especially the Google Earth Engine platform provides a lot of precalculated datasets from various sources such as MODIS [Did21], however each index is separately available, therefore requiring a lot of manual 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.

In comparison to MODIS, Sentinel satellite data provides a significantly higher resolution at 10 - 60 m^2 per pixel compared to 500-1000 m^2 per pixel for MODIS but there are no precalculated indexes available for Sentinel data on the Google Earth Engine 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 [23].

In comparison to MODIS and Sentinel, many LiDAR datasets are closed-source and are

[22] `https://gcos.wmo.int/en/essential-climate-variables/table`, *last accessed: 08.08.2023*

[23] `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
Vegetation Index		**Radiation Index**	
Normalized 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	**Building Index**	
Density of Low Vegetation	LiDAR	Normalized Difference Buit-Up Index (NDBI)	Landsat 8
Density of Medium Vegetation	LiDAR	Urban Index (UI)	Landsat 8
Density of High Vegetation	LiDAR	Index-based Built-Up Index (IBI)	Landsat 8
Water Presence Index		Building Density	LiDAR
Modified Normalized Difference Water Index (MNDWI)	Landsat 8	**Urban Morphology**	
Normalized Difference Water Index (NDWI)	Landsat 8	Sky View Factor	LiDAR
Bare Soil Index		Standard Deviation (STD) of Building Height	Local Authority
Normalized Difference Bareness Index (NDBaI)	Landsat 8	**Moisture Index**	
Bare Soil Index (BI)	Landsat 8	Tasseled Cap Transformation Index	Landsat 8
Enhanced Built-Up and Bareness Index (EBBI)	Landsat 8	Normalized Difference Moisture Index (NDMI)	Landsat 8
Density of Bare Soil	LiDAR		

Table 4.3: Indexes used by Alonso and Renard [AR20] to predict air temperature.

not available for research purposes. This is unfortunate as LiDAR data provides a very high resolution of 5-10 cm^2 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 sky view factor which seems to have a significant impact on air temperature modlling [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:

- Topographic
  - Slope (°)
  - 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 air temperature mapping study done in Oslo by Venter et al. [VBEM20], the following data was used:

Hyperlocal Mapping in Oslo: Red Landsat 7, 8 and Sentinel 2 Open source L: 30 m, S: 10 m Green Blue Near infrared Short-wave infrared 1 L: 30 m, S: 20 m Short-wave infrared 2 NDVI L: 30 m, S: 10 m IBI L: 30 m, S: 20 m Land surface temperature Landsat 7, 8 30 m Elevation above sea STRM 30 m 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 4–20 m Building height sd 20–100 m Fractional tree cover LiDAR + orthophoto Tree height Distance to coast Global water occurrence Open source 30 m Distance to fresh water

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

Table 4.4: Features for Air Temperature Interpolation Used in this Work

Features used in this Work

Air temperature, relative humidity, atmospheric pressure, precipitation, and wind was sourced from Netatmo and Sensor.Community 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 Earth Engine [GHD+17] as it offers a unified way of accessing data and offers anhanced 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 Earth Engine 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:

- MODIS_061_MOD15A2H: MODIS Leaf Area IndexFPAR (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 incoporated into the models either by longitude and latitude values in coordinate rerence system EPSG:4326 or by calculating distances between locations in meters. The features used in this work are shown in Table 4.4 and were chosen based on availability and relevance for air temperature modelling. The amount of features in the initial scope of this work is quite limited and could be increased to gain better prediction results.

5 Evaluation

The goal of this chapter is to evaluate different use-cases for ML-based interpolation in the context of air temperature 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 feasability of two main use cases:

1. Interpolation of air temperature for a specific location
2. Areal interpolation of air temperature

Next to these main uses cases we discuss in this work, the different regression models can be used in many other ways, primarily based on the features used during training. Some regression models could for example be used to predict future air temperature 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+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

In order 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, tho 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 (baseline), KNN, Random Forest, SVM, and Histogram-based Gradient Boosting, with all features against each other to get an understanding of the overall performance of the models and maximum achivable 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 in order 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 Random Forest Regressor and the number of iterations for the Histogram-based Gradient Boosting Regressor, which are both increased from 100 to 200 to improve the performance of the models as found out by previous exploration. In order to get the best possible performance of each model, an exhaustive grid search would be prefered for all models in order 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, RSME, R2. MSE is the most commonly used error metric and is more sensitive to outliers, whereas MAE punishes outliers less but is therefore also more variable. Related work commonly uses RSME as it has the same unit as the response variable, therefore RSME is used as the main error metric. R2 is additonally 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 air temperature 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 air temperature can be interpolated more easily, especially over a longer period of time. 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 air temperature 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 preprocessing for the specific models. Only Histogram-based Gradient Boosting 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, not column. All input data is normalized using a standard scaler. This preprocessing 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.

4. Choose the most promising model and further evaluate the influence of different features, e.g. distances between neighbours, and the amount of training data and time intervals. This step only selects a subset of stations of particular interest, e.g. in the city center or near water, adds 5-Fold Cross Validation and grid search to fine-tune the hyperparameters of the model.

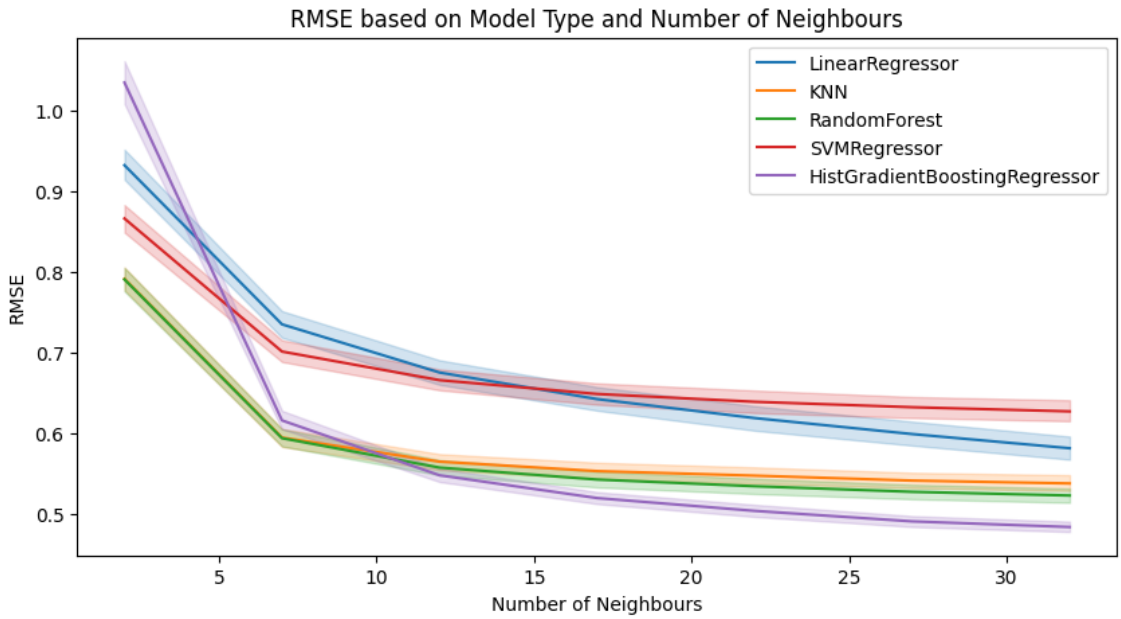

Figure 5.1: RMSE by Model Type with the Confidence Interval of 95%

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 accross both Hamburg and Stuttgart for the month of June 2023. For all models, one evaluation was done using the imputed dataset without missing values, as Linear Regression, KNN, Random Forest and SVM Regression cannot

handle missing values. Gradient Bootsting 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 Random Forest seem to hit a plato 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 (TODO: Link), 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 1.

5.1.2 Further Evaluation - Gradient Boosting

HistGradientBoostingRegressor without imputation achives overall the lowest RMSE starting with more than 10 neighbours and continously improves with more neighbours, however the improvements after 30 neighbours are really small. This model is great because it has native NaN support and we can save the imputation step that can possible introduce bias into the model. The difference to KNN and Random Forest is not that big with a little less than RSME 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 Random Forests 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 hypotheses is, that closer stations have a higher influence on the prediction quality, e.g. if closer stations are choosen as neighbours the prediction quality is better compared than if the same number of neighbours is choosen 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 to close to the station.
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

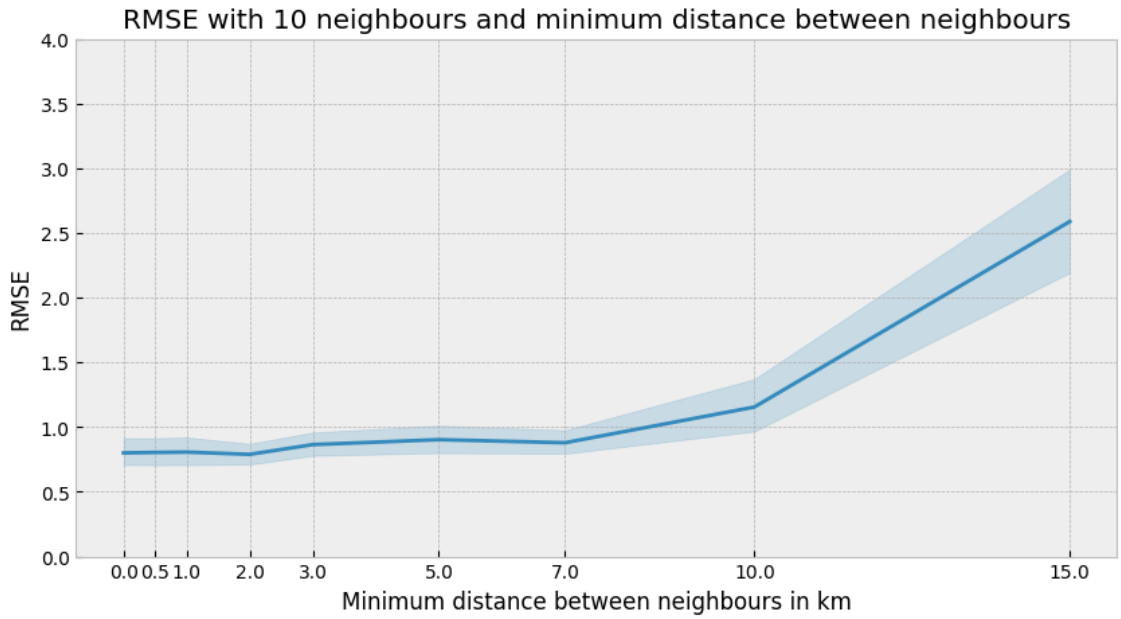

Figure 5.2: RMSE for Increasing Minimum Distance with 10 Neighbours, Hamburg, Netatmo

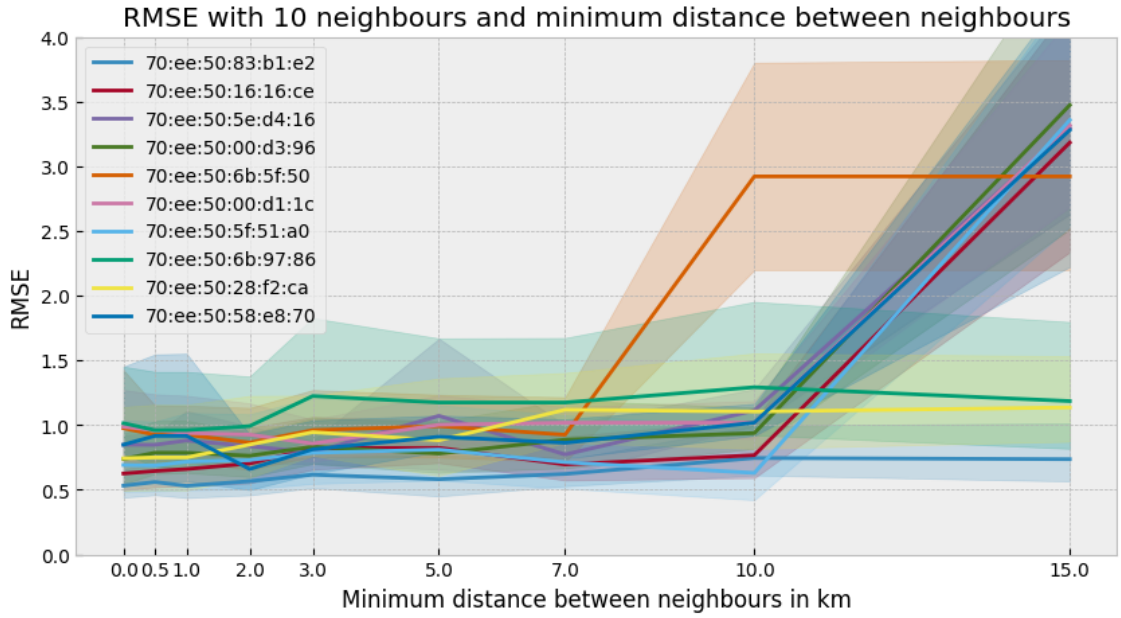

Figure 5.3: RMSE for Increasing Minimum Distance with 10 Neighbours By Station Id, Hamburg, Netatmo

and different climatic conditions, e.g. distance to water, situated in the city center etc. The list of stations can be found in Appendix 2. The number of neighbours was set to 10 as this number was the number at which Gradient Boosting putperformed 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 kilometers 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... ta_n, time, humidity_1... humidity_n, pressure_1... 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 importances.

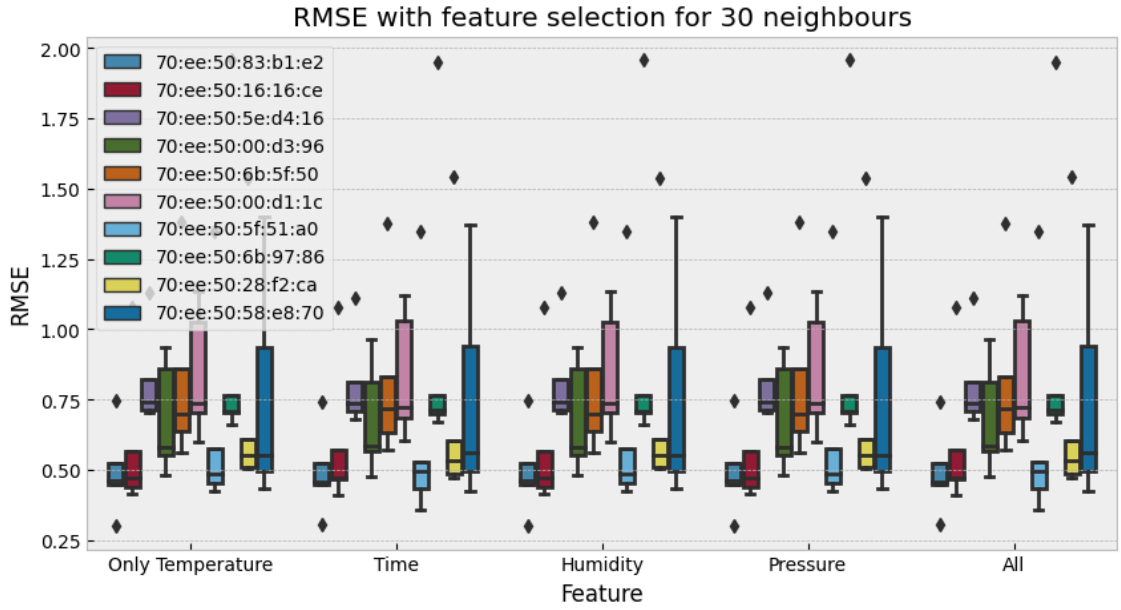

Figure 5.4: RMSE based on Features Selected with 30 Neighbours, Hamburg, Netatmo

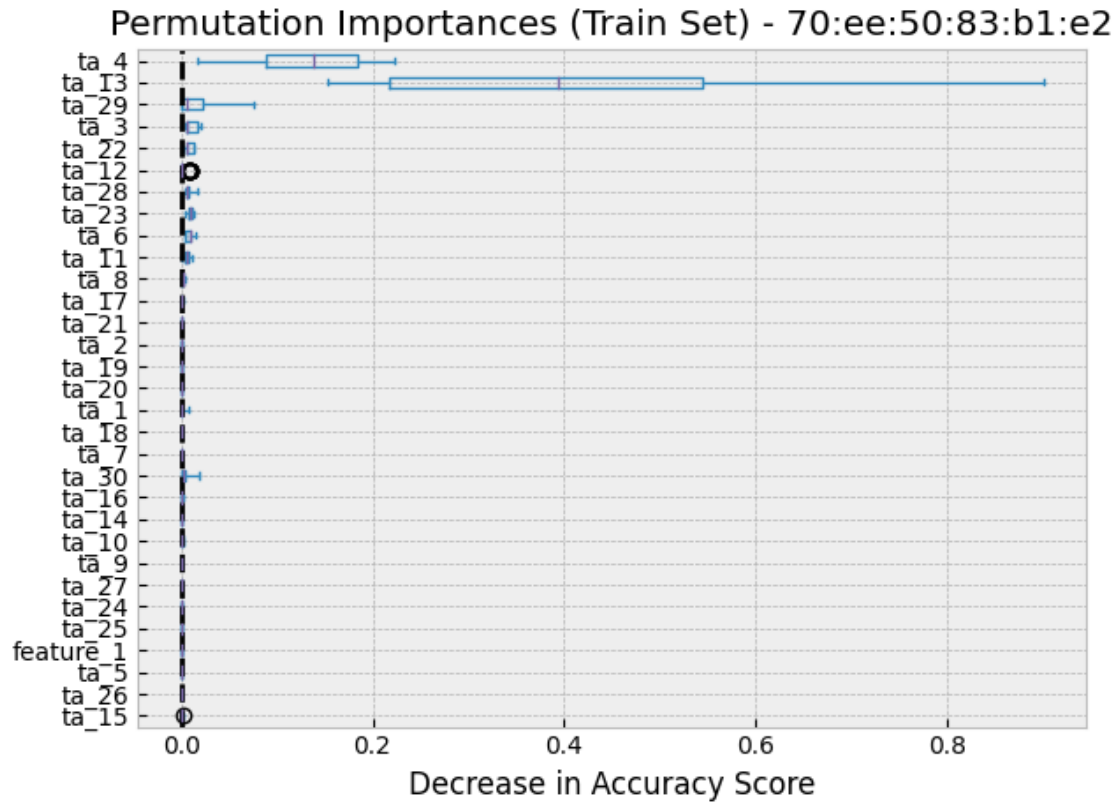

Figure 5.5: Permutation Importance for Single Station on 5-Fold Cross Validation, Hamburg, Netatmo

Influence of Quality Control, 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. Related studies in the field of sensor networks suggest, that stationary sensor networks show less variability, but suffer from lower spatial coverage. In comparison, fully mobile sensor networks have a higher spatial coverage, but suffer from higher variability. Combining both approaches could therefore be a promising approach to increase the spatial coverage of the sensor network, while still maintaining a high prediction quality.
In the context of air temperature sensing, air temperature sensors could be mounted to busses, cars, scooters, or bikes. Unfortunately, there are currently no such datasets 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, especially for Sensor.Community in the m5 buddy check. In order to further investigate the influence of missing data, we instead of using data that passed the m5 check, we will use data that passed the m4 check, which is less strict and therefore has more data available. An interesting exploration could be to use the previous sensor with Netatmo Station Id *70ee:50:83:b1:e2* and choose on of the neighbours that has a high influence on the prediction quality, e.g. neighbour 4, 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 one but multiple neighbours are turned into simulated moving sensors, maybe with different intervals.
First, we need to get an understanding how the different QC steps influence the interpolation process. Therefore, we compare the RMSE for both m4 and m5 QC step for the 10 already used stations with 30 neighbours and the time feature.

5.2 Areal Interpolation of Air Temperature

The second use-case for ML-based interpolation of air temperature is the areal interpolation, e.g. turning a set of single data points into a continuous temperature 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, sky view factor, solar radiance, etc. have similar air temperature. The main challenge here is to find the right features that capture the dependency between the different locations.
In the first step, we simply use the coordinates of each data point in order to train the

model in addition to remote sensing data, as land coverage and surface roughness have a high impact on air temperature.(Cite LCZ studies). Further improvements

- parameters: - grid cell size

- todos: - try to get surface temperature for specific time step -> LST high influence on TA?

5.2.1 Model Comparison

- more features compared to location interpolation: land coverage, soil temperature, sky view factor, from satellite data and google earth engine

- improvements: - predict air temperature based on LST for better training

baseline: mean, IWD, ordinary kriging (with different kernels)

5.2.2 Geostatical Interpolation Baseline

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 inverse weight distance (IWD) or k-nearest neighbours (KNN) that are easy and performant, but struggle to capture more complex interdependencies, there are also more complex methods available. The most common geostatistical method for interpolation is Krigin, which is based on a gaussian process and

In the scope of this work, we unfortunately cannot compare all of these methods with each other and therefore need to focus on a subset of methods. EBK and EBKRP are one of the most commonly used methods for temperature interpolation (cite). According to [NAEB23], EBKRP continously outperforms EBK in different weather station density scenarios, therefore we will use EBKRP as a baseline for our comparison.
In the following, the machine learning fundamentals for this work are explained and the different ML regression model types are introduced.

5.2.3 Further Evaluation - HistGradientBoostingRegressor

5.2.4 Datasets for Evaluation

As a first step for training and testing areal air temperature interpolation methods, we need to define the test area and the data we use for training. In this context, it could be interesting to compare different sensor network providers. For this use-case we have several test areas that we could validate. To get an overview of the sensors left after QC for Sensor.Community, Table

-> could also compare with Hamburg (closer to water, more wind)

- find locations that are: - somewhat near reference stations for comparison - high station density - interesting features (e.g. parks, water, ...)

- locations in Germany for Sensor.Community are: - Stuttgart (birthplace of Sensor.Community) with high station density + one reference station somewhat close - Hamburg with lower sensor density, but 2 reference stations (interesting to see changes between those over realtively short distances) and close to water/river (Elbe) - other candidates (but with way fewer sensor and no reference stations): Cologne, Munich, Freiburg (interesting that they have an airport but no reference station), Lüneburg - interesting to note, that Berlin did not have any sensor stations that passed the QC step

- we choose stations from Stuttgart and Hamburg for the evaluation -> identify stations with high mean daily temperature difference to reference station (to see if they are in a heat island)

Stuttgart

Station Locations:

DWD Reference location: 48.6883;9.2235 -> DWD id: 4931, Daily Max, Mean and Min values for June 2023

Hamburg

Todo: add Hamburg station locations

TODO First compare only air temperature for stations that are near a reference station and have enough buddies for sensor community - only for june 2023 from sensor community, not january as heat islands are not as important in winter (which depends on the context ofc if the goal is f.e. to save heating costs, could be a factor)

Steps in the evaluation: - find sensor community stations near reference stations that passed m5 QC step - Compare different interpolation approaches - linear, forests, KNN, NNetwork - with different amounts of data available (99% -> 1%) and see how RSME evolves

Steps: - deterministic approaches - "naive" approach with nearest neighbor -> show high error rate - probabilistic approaches - reference approach with geostatical methods (ordinary kriging, empirical bayesian kriging, EBK with regression) -> still high error rate, especially with lower density of weather stations/bad support for irregularly spaced data - ordinary krigin: - temperature semivariogram first - add additional features (e.g. soil temperature, land coverage, sky view factor, ...) as semivariograms and use cokrigin to combine them - empirical bayesian kriging: - not implemented out of the box (pykrige)

- semivariogram: https://pro.arcgis.com/en/pro-app/latest/help/analysis/geostatistical-analyst/understanding-a-semivariogram-the-range-sill-and-nugget.htm

- deep learning approach with neural networks -> itteratively improve model by adding additional features, compare with reference approach

1. Data Collection and rough analysis 1.1. Get data from Sensor.Community 1.2. Get data from DWD 1.3. Get data from satellites via Google Earth Engine 1.4. Get data from Netatmo (Hamburg, try to get Stuttgart as reference with historical data via API) 2. QC steps (TA first) 2.1. Try to remove indoor stations and stations with irregular data 2.2. Remove stations based on buddy check (removes a lot of sensor community stations due to lower density) 2.3. Reference station has high quality data already, but can be used as a reference (interesting the comparison from 2m TA to 5cm TA) 2.4. Discuss QA for other parameters (Humidity, Pressure etc.) 3. Comparison of models 3.1. Simple models with only coordinates as input features and TA as target feature 3.2. Add additional features (land coverage (NDVI, EVI)) and see how they improve the model 3.2.1. For land coverage etc. compare different satellites and resolutions 3.2.2. For coordinates/distances, compare different ways of calculating distances or modelling them in the model 3.2.3. Compare influence of normalization 3.3. Try to create a more complex NN model to show capabilities of NNs and try out LSTM for time series 3.4. Choose final candidate that seems the most promissing and compare with reference approach 4. Comparison with refrence approaches 4.1. Compare with simple Ordinary Kriging approach (with different kernels) 4.2. Discuss problem of not being able to extrapolate

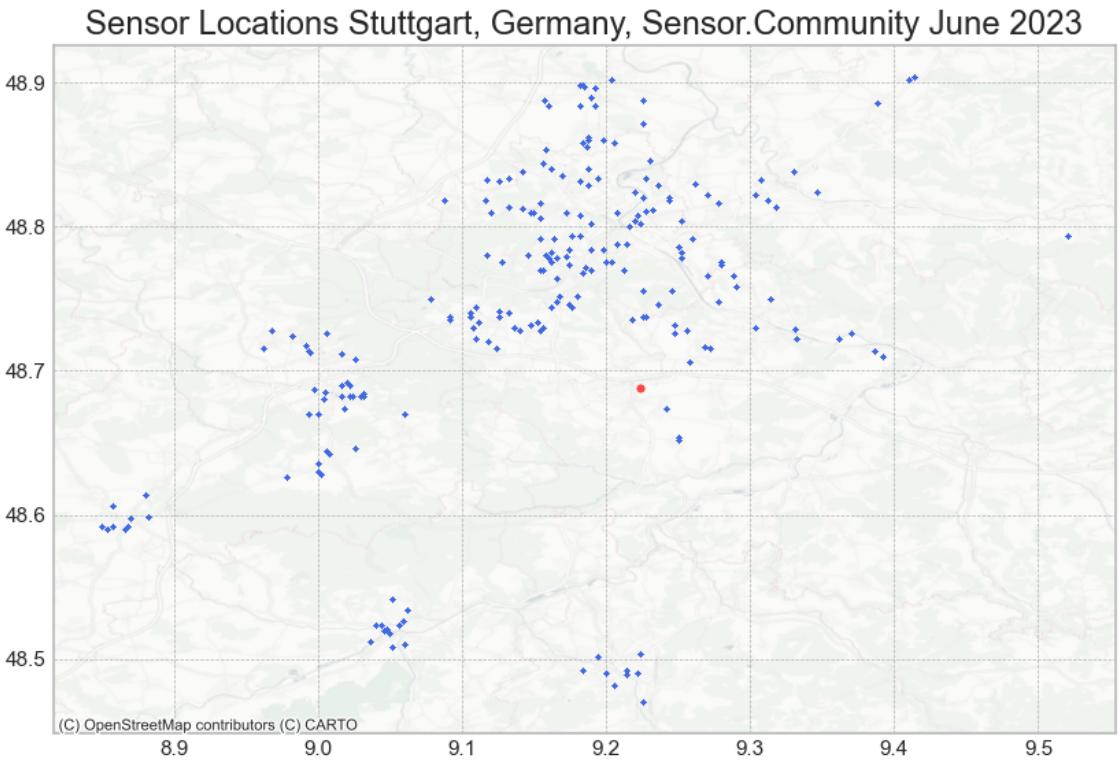

Figure 5.6: Locations for Sensor.Community around Stuttgart, Germany, June 2023

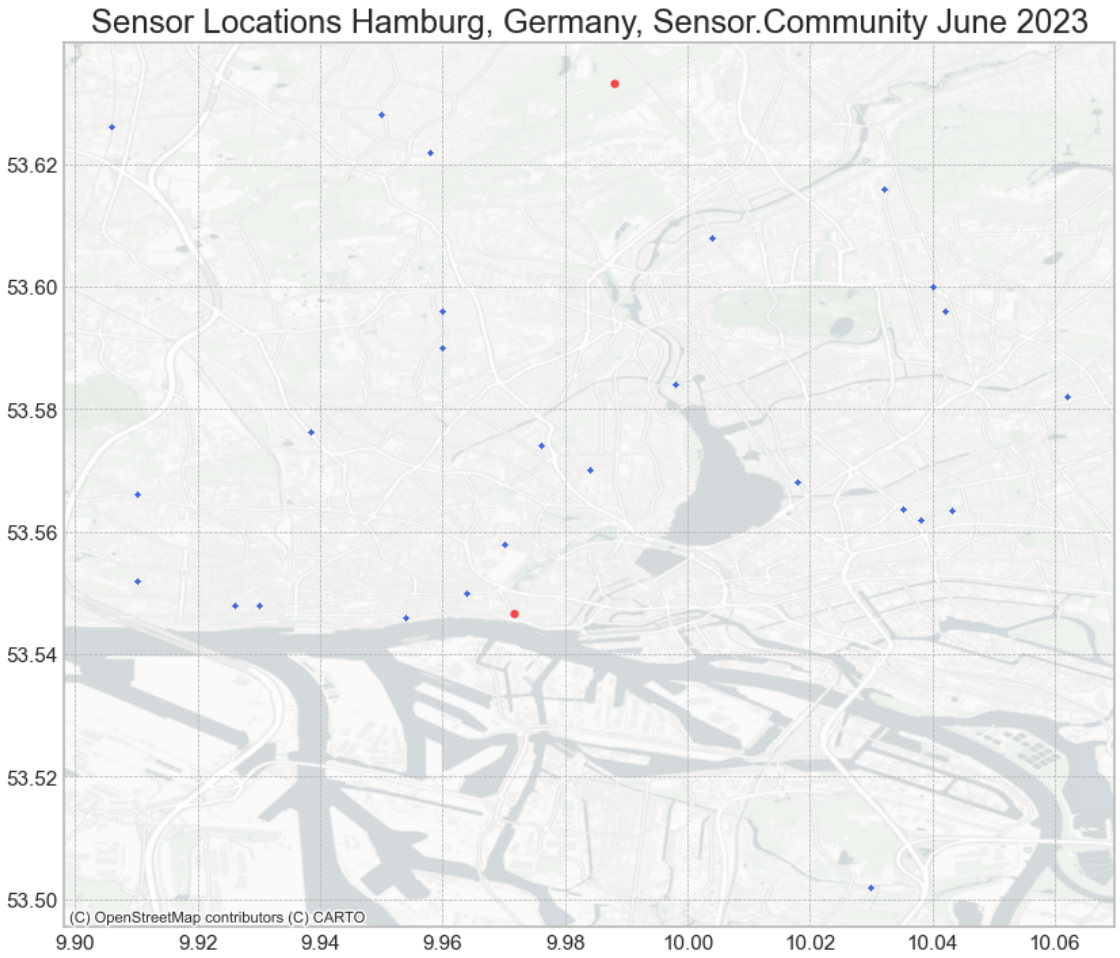

Figure 5.7: Locations for Sensor.Community in Hamburg, Germany, June 2023

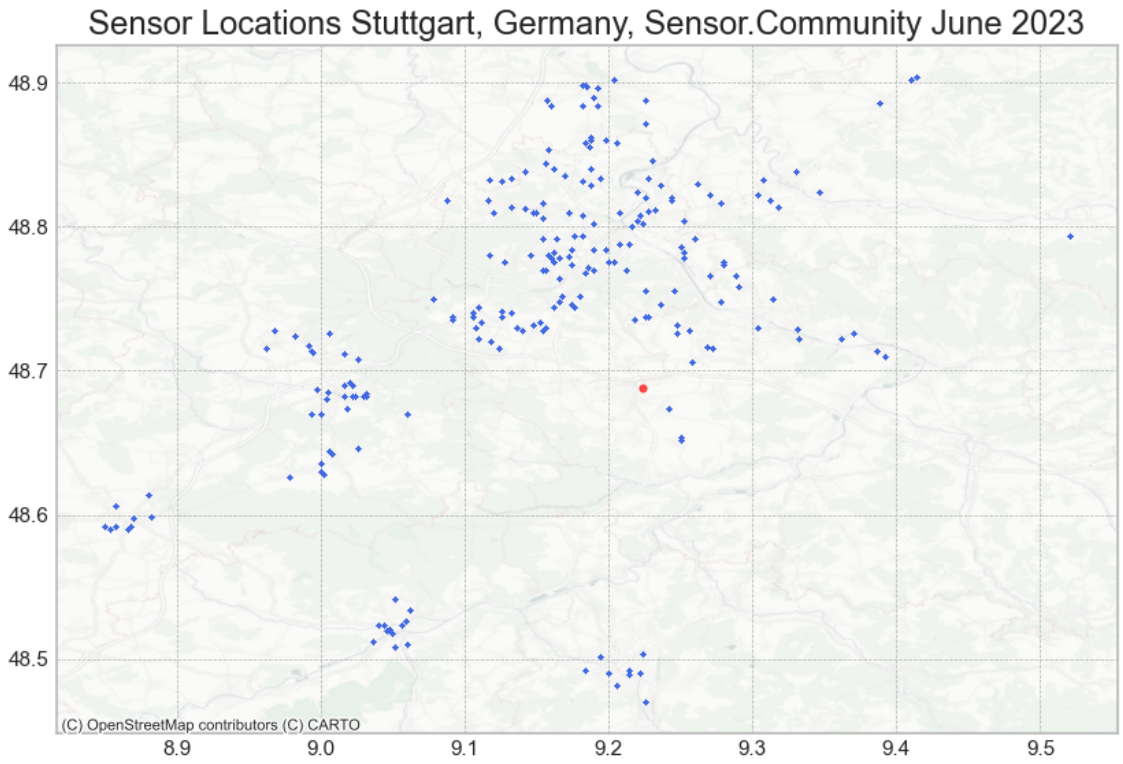

Figure 5.8: Locations for Sensor.Community around Stuttgart, Germany, June 2023

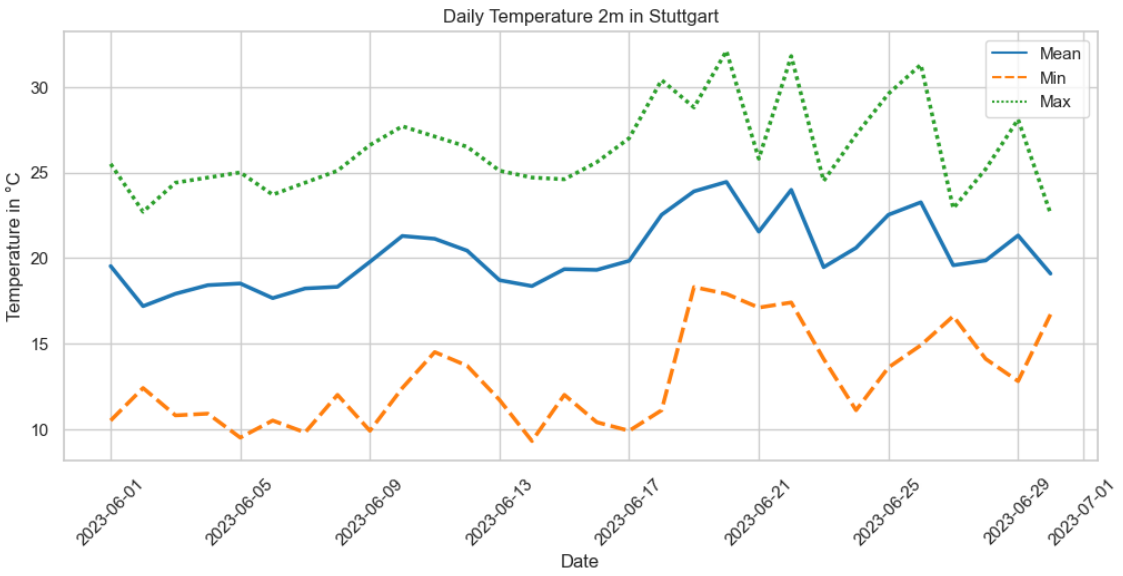

Figure 5.9: Daily Mean, Max, and Min Air Temperature at 2m in Stuttgart, Germany, June 2023, DWD Station 4931

6 Conclusion

single station interpolation works very good with RMSE between 0.4 and 0.5 - because other factors, such as height of the station due not play an important part, as long as the distribution of the temperature is the same

areal interpolation way harder. probably because information is missing (target variable for location/grid cell and height), making it hard to f.e. select stations only at 2m height that are representative for a larger area. Combination with LST however possible (should be investigated)

how these could be integrated -> interpolation service / temperature map that integrates multiple providers and allows interactive analysis. current tool set with python very flexible, but slow. dataset availability could be way better, especially with preprocessed data, so that one can focus on the actual comparison.

comparison of model should have been done with more models, and f.e. should include state of the art implementation from f.e. ArcGIS.

final recommendation. create virtual sensors based on moving sensors -> use those virtual sensors to augment actual sensor data and then use eact interpolation methods such as Kriging to create a smooth surface.

6.1 Summary

- discuss low amount of parameters and curse of dimensionality

TODO: Rewrite once Evaluation done

With the growing need to analyze microclimates of cities in order to protect them against new phenomena like UHIs, this paper proposes the use of citizen-owned sensor networks that offer a higher spatial and temporal resolution of data points in comparison to traditional approaches such as relying on LST data. This approach also comes with challenges such as observing areas without stationary sensors and identifying poor quality data-points from broken or incorrectly installed sensors. With the obtained data, we create a temperature interpolation service that predicts temperature data between data points using a regression model that can act as a building block for further temperature-based analysis by abstracting the underlying single data-points in the data-layer away.

In order to improve the quality of temperature predictions for unobserved areas, we investigate how temporary sensor readings, f.e. by attaching sensors to buses, bikes or e-scooters, can be used to capture temporary local meteorological snapshots, and how these snapshots can be incorporated into the interpolation process. We also evaluate which features need to be captured to generate the most accurate predictions, if machine-learning is a suitable approach, and how it compares to more traditional geostatistical approaches.

Possible improvements: - QC -> use different time intervals, investigate different qc processes for other variables, how it looks with reference stations, combining more data, bias of specific sensors etc., not default parameters, e.g 3000m radius too big?

Support for existing findings: - harder to interpolate when it's hotter/air temperature is more variable - harder to interpolate when there is less data - surface temperature/temperature near the ground way more variable and less useful in predicting UHIs

Other options (not yet shown): (harder to interpolate closer to water/big rivers)

New findings: - Sensor.Community data cannot substitute Netatmo data due to low sensor density, and data quality an issue, however data availability is comandable

6.2 Future Outlook

TODO

Bibliography

[Alt92] ALTMAN, Naomi S.: An introduction to kernel and nearest-neighbor nonparametric 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, Mehmethan ; 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+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+17] CASTELL, Nuria ; DAUGE, Franck R. ; SCHNEIDER, Philipp ; VOGT, 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

[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+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+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+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+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

[HDJ+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+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+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+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+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+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+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+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+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+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+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/PyKrige: 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+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+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+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

[sen] *Sentinel-2 MSI: MultiSpectral Instrument, Level-2A*. `https://developers.google.com/earth-engine/datasets/catalog/COPERNICUS_S2_SR`,

[SHN+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+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+20] SEKULIĆ, Aleksandar ; KILIBARDA, Milan ; HEUVELINK, Gerard B. ; NIKOLIĆ, Mladen ; BAJAT, Branislav: Random forest spatial interpolation. In: *Remote Sensing* 12 (2020), Nr. 10, S. 1687

[SKP+20] SEKULIĆ, Aleksandar ; KILIBARDA, 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+22] THOPPILAN, Romal ; DE FREITAS, Daniel ; HALL, Jamie ; SHAZEER, 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: Hyperlocal mapping of urban air temperature using remote sensing and crowdsourced 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)

[WYC+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+20] YUAN, Qiangqiang ; SHEN, Huanfeng ; LI, Tongwen ; LI, Zhiwei ; LI, Shuwen ; JIANG, Yun ; XU, Hongzhang ; TAN, Weiwei ; YANG, Qianqian ; 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 Sklearn Machine Learning Model Parameters for Single Station Interpolation

Listing 1: Random Forest Regressor Parameters

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

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

Listing 2: Histogram-based Gradient Boosting Parameters

```
class sklearn.ensemble.HistGradientBoostingRegressor
(loss='squared_error', *, quantile=None, learning_rate=0.1,
max_iter=200, max_leaf_nodes=31, max_depth=None, min_samples_leaf=20,
l2_regularization=0.0, max_bins=255, categorical_features=None,
monotonic_cst=None, interaction_cst=None, warm_start=False,
early_stopping='auto', scoring='loss', validation_fraction=0.1,
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 interations yield the same result.

2 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 1.

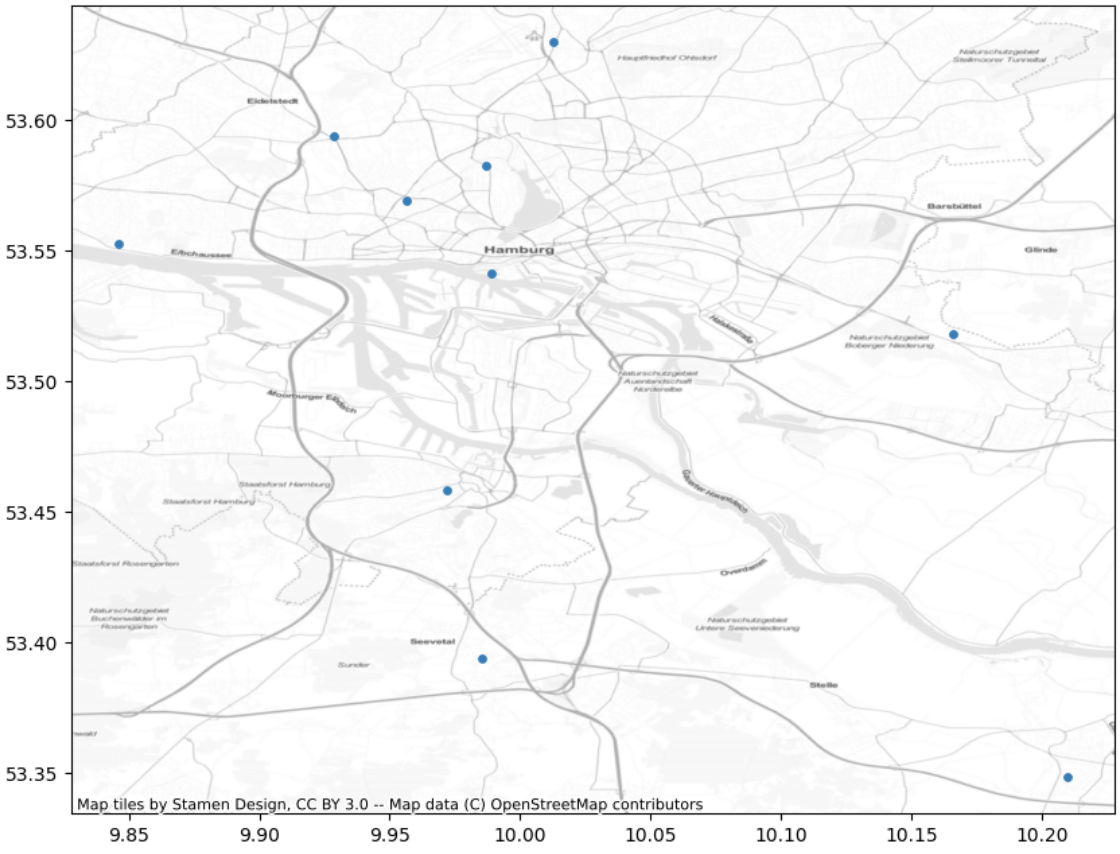

Figure 1: Netatmo Stations for Minimum Distance Between Stations, Hamburg

3 Sensor Community

Listing 3: Sensor Distribution by Country


```
{
    "2023-06-24": {
        "bme280": {
            "WORLD": "5006",
            "DE": "1558"
        },
        "bmp180": {
            "WORLD": "159",
            "DE": "72"
        },
        "bmp280": {
            "WORLD": "254",
            "DE": "100"
        },
        "dht22": {
            "WORLD": "5292",
            "DE": "2590"
        },
        "ds18b20": {
            "WORLD": "29",
            "DE": "11"
        },
        "hpm": {
            "WORLD": "6",
            "DE": "1"
        },
        "htu21d": {
            "WORLD": "105",
            "DE": "14"
        },
        "laerm": {
            "WORLD": "233",
            "DE": "117"
        },
        "nextpm": {
            "WORLD": "1",
            "DE": "1"
        },
        "pms1003": {
            "WORLD": "7",
            "DE": "2"
        },
```

```
        "pms3003": {
            "WORLD": "7"
        },
        "pms5003": {
            "WORLD": "255",
            "DE": "16"
        },
        "pms6003": {
            "WORLD": "1",
            "DE": "1"
        },
        "pms7003": {
            "WORLD": "206",
            "DE": "8"
        },
        "ppd42ns": {
            "WORLD": "2",
            "DE": "1"
        },
        "radiation_sbm-19": {
            "WORLD": "3"
        },
        "radiation_sbm-20": {
            "WORLD": "7"
        },
        "radiation_si22g": {
            "WORLD": "71",
            "DE": "54"
        },
        "scd30": {
            "WORLD": "2"
        },
        "sds011": {
            "WORLD": "12199",
            "DE": "4964"
        },
        "sht15": {
            "WORLD": "1",
            "DE": "1"
        },
        "sht30": {
            "WORLD": "130",
            "DE": "17"
        },
        "sht31": {
            "WORLD": "259",
```

```
            "DE": "57"
        },
        "sht35": {
            "WORLD": "11",
            "DE": "5"
        },
        "sht85": {
            "WORLD": "2",
            "DE": "2"
        },
        "sps30": {
            "WORLD": "279",
            "DE": "53"
        }
    }
}
```

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit im Masterstudiengang Wirtschaftsinformatik selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen 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: _____