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

Abgabe: 08.2023

A distributed system is one where the failure of some
computer I've never heard of can keep me from getting my work done.
– *Leslie Lamport*

Abstract

Many science-based applications need continuous or gridded input data in order to work properly. This paper investigates how single data-points can be combined to create a continuous data layer, in which missing data points are interpolated. An example for such an application would be the prediction of temperatures coming from single sensors and weather-stations, that can be combined to detect Urban Heat Island (UHI)s. UHIs are weather phenomena that get ampflified among other things by the ongoing densification of urban areas to create more living space, typically accompanied by the removal of green areas that can help with the dissipation of heat. These heat islands, in which the temperature is significantly higher than in surrounding areas, pose a threat to the health of the urban population, especially to the elderly, children and people with existing health conditions. Traditionally, UHIs are detected using Land Surface Temperatures (LST) captured by satellites, that usually have the downside of low spatial and temporal density. This paper proposes an alternative approach by creating a machine-learning model that is able to interpolate missing data between data-points coming from citizen-owned sensor networks that are combined with mobile sensors, which can be attached to rental bikes, buses or e-scooters, to gain temporary insights into otherwise unobserved areas. The model combines data streams of sensor readings with historic data and creates a fine-granular continuous data-layer, in this case for temperature, which allows for an accurate localization of UHIs.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

LST	Land Surface Temperatures
UHI	Urban Heat Island

1 Introduction

1.1 Motivation

In 2023 56% of the human population already lives in urban areas with the number projected to continiously increase to 68% by 2050 [1]. Combined with the ongoing climate change and urban densification, 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. A recent phenomenon is the appearance of so called urban heat-islands (UHI). A heat-island is a local occurrence of significantly higher temperatures than surrounding areas that pose a health risk, especially for the elderly, children or citizen with prior health-issues [MBG15].
In order to detect UHIs, Land Surface Temperature (LST) is commonly used. While allowing for a cheap 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]. This spatial and temporal resolution, of f.e. spatial resolution of 0.01° longitude and 0.01° latitude (equal to roughly 1.11km by 1.11km) and temporal resolution of monthly average surface temperature as offered by LST data provided by the European Space Agency (ESA) Climate Office's data set [GVP], is not enough to effectively analyze the urban microclimate. Another candidate that comes to mind are weather stations. They usually provide hourly, for current values sometimes even 10 min interval readings of temperature, rain and wind, but don't offer the necessary spatial resolution. Lastly, there is the possibility of deploying sensor networks to closely monitor the climate of the city, but this approach can be quite cost intensive for a large amount of sensors over a long time period [CMY+15]. An alternative that is less costly would be to instead rely on citizen-owned sensor networks from the existing Smart Home and Internet of Things (IoT) infrastructure, like Sensor.Community [2] and Netatmo [3], which offer a temporal resolution of 5 min for temperature and wind, hourly for rain, while also having a comparably high spatial resolution. This approach has the desired temporal resolution and has been shown to

[1] https://ourworldindata.org/urbanization#by-2050-more-than-two-thirds-of-the-world-will-live-in-urban-areas
[2] https://deutschland.maps.sensor.community/
[3] https://weathermap.netatmo.com/

work well in [MFG+17], but there might be areas, such as industrial zones, where citizens are not able or not allowed to install their personal sensors. In order to also gain insights in such previously unobserved areas, we propose the usage of mobile sensors that could be installed on buses, bikes or e-scooters to gain temporary snapshots and improve the spatial resolution even further. As research realted activities commonly rely on continuous or gridded data fields, there needs to be a way to convert these single data points from the different sensors into a continuous data-layer.
In this paper, we propose a solution to this problem by training a machine learning regression model, that allows for the interpolation of missing data-points. Based on sensor readings, from the sensor networks and mobile sensors, of commonly collected weather information, such as temperature, humidity, rain, pressure, wind, and possibly other variables such as vegetation indexes [AR20], the model then creates a continuous data-layer that allows for a holistic view of the observed variable, in this case temperature.

To do (??)

1.2 Objective

current situation: abundance of data (data quality unknown), but different data sources at different places with different formats, making it hard to work with different sources at the same time - smart city example -> heat island detection - sensor networks (statinary + mobile) - LST satellite data (lack of spatiotemporal resolution, not suited for microclimate) - vegetation indexes (in geoinformation systems/portals) - etc. - currently these sources are used independently from each other, but how can they be integrated? - hybrid approaches have shown that combining different approaches (smart city stationary + moving sensors) to give better prediction quality than singular approach (reasons: unobserved areas) - statistical models offer not enough flexibility/are to cumbersom to work with (and probabilities are not known) - ML is a good fit to analyse patterns and rules - but effort to retrain models for each application expensive - how can ML be used to integrate the different types of data? - how can data availibility of contexttual data be improved by leveraging ML approaches for different topics? - for the heat map detection, what you need is a fine-granular temperature map, that allows for outlier detection - either train your own model on a huge dataset - or access a temperature map -> this will be discussed, how to use ML for this and improve data availibilty by using interpolation - additonal topics to briefly discuss (mainly apply by implementing ML models) - what work needs to be done before using the data in ML (preprocessing, transformation, outlier detection etc.) - one way of preparing data is to interpolate missing data to create continous gridded features (focus of this work) - how ML can we improve the interpolation quality of features? -> turn sparse features into denser versions with interpolation, interpolation based on other features present

Methodology

In order to validate these things, we need high quality data-sets with many different features, but there are not many available, even tho for certain things (atmosphere etc.) there are OpenData sets available. Goal is: implement a temperature map based on ML that interpolates missing data based on historical and current data. Compare prediction quality of different type of models (DL, random forests etc.) based on cross-validation (simulate missing data by leaving data points out on purpose) -> experiment with different intervals, densities etc. -> could also compare to Krigin (as a typical geostatistical approach for temperature interpolation) Subgoal: In order to achive this, we also need high quality data-sets to train and validate the model -> source them from OpenData platforms

1.3 Structure of this work

Research methods:
- literature research as foundation
- heat island detection - smart cities -> sensor networks vs LST - what type of other data is available? geoinformation, vegetatition, for micro-climate: shades of bigger buildings?
- - prototyping
- implement pipeline to pre-process different types of data
- feature extraction
- implement ML model
- deploy ML model

- create/search for fine granular data sets
- add new contextual data to exitsing data sets
- discuss different types of data (gridded vs continous vs data points) and methods for each
- train ML models with different methods (deep learning, random forests etc.) and different features enabled

- cross validation of results
- discuss validation techniques and indicators (RSME, MSE)

The rest of the thesis is structured as follows. Chapter 2 begins with an analysis of related work, where important literature is discussed, that forms the foundation of this research. In chapter 3, the focus lies on describing the service architecture, that shows how a machine learning model can be deployed in different contexts to improve data availability. Which machine learning approaches can be used to interpolate missing data and how they differ from each other is discussed in chapter 4. In Chapter 6, the different

ML approaches are compared and cross validated with each other based on the different model that are trained on the obtained data-sets. Finally, chapter 7 discusses the findings of this thesis and gives an outlook into future work and research directions.

2 Related Work

In the following chapter we lay the foundation for the research conducated in this thesis. We start off with an introduction of the topic of urban heat island (UHI) detection, which is one of the motivating factors behind this work. UHI research is part of the discipline of modern climatology and relies on many different sub-disciplines like meteorology, thermal dynamics, geology and many more. After discussing traditional ways to detect UHIs, such as using remote satellite data, we discuss newer approaches, especially in the context of smart cities, that focus on collecting air temperatures in urban environments. Measuring data in the urban environment comes with many challenges [Oke06], that can lead to missing or wrong observations for given places. In order to improve data availibility in the urban climate context we compare traditional interpolation techniques based on (geo-)statistics with more recent machine-learning based approaches, that could be used in this highly complex urban setting. This interpolated data could be used to improve interpolation or prediction models and give additional insights into the influences of urban environments, f.e. in the context of UHIs and their detection and analysis. Finally, in order to compare traditional geostatistical models with ML-based models, we need comprehensive data-sets of urban climate data. An overview of available data sources is given in section **??** together with an introduction to the OpenData movement.

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 which 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.

Shared UHI Challenges

Some shared challenges are: 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 (cite), 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,... (cite, todo find the influence factors). Additionally, the urban climate is also influenced by surrounding regional/meso-scale climate phenomena such as storms, valleys, mountains, large waterbodies, costlines and more (cite).

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 to 2.3 show a detailed view of the meso-, local- and micro-scale of the urban climate, as illustrated by Oke 2006 [Oke06].

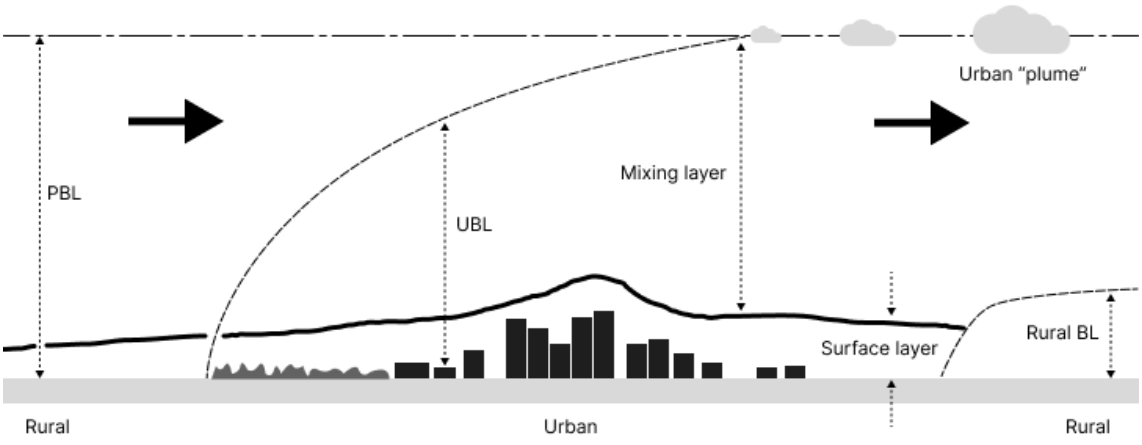

Figure 2.1: Mesoscale view of the urban climate, redrawn from [Oke06]

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.

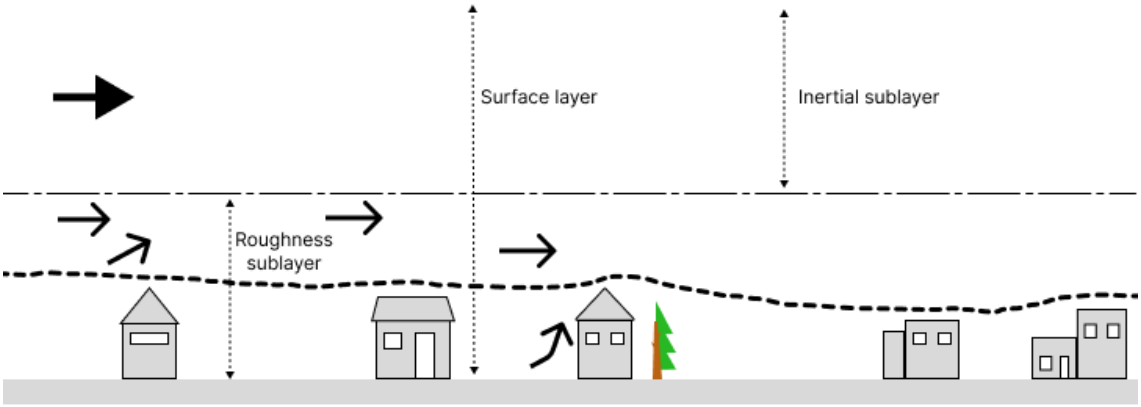

Figure 2.2: Localscale view of the urban climate, redrawn from [Oke06], (Todo finish)

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. Todo: more infos

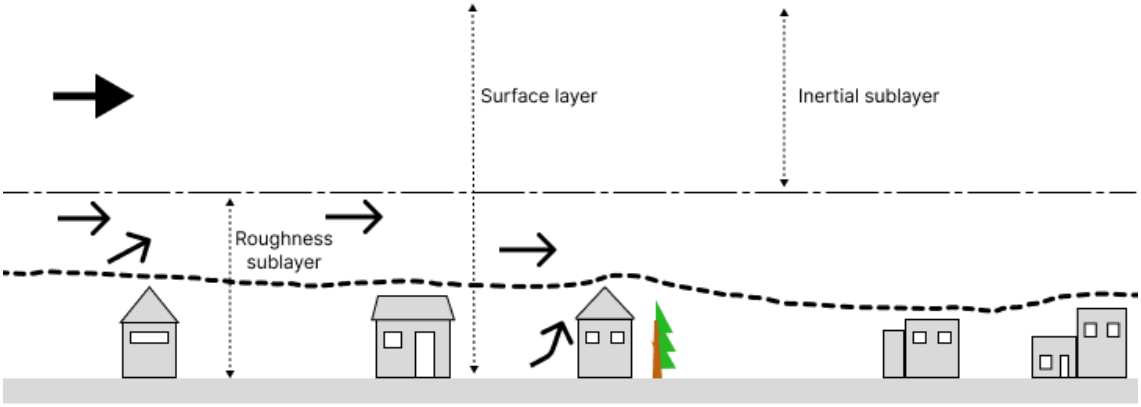

Figure 2.3: Microscale view of the urban climate, redrawn from [Oke06], (Todo)

The microscale deals with the characteristics of each individual surface area. On of the main measurements on this layer is the surface temperature, which is primarily influenced by solar radiation. It is important to note, that surface and air temperatures are correlated, but this correlation varies greatly based on other surrounding influences such

as wind velocity or humidity [SB92]. todo: more

Vertically UHIs can be devided into three mayor types [Oke76, OMCV17], namely Boundary Layer Heat Island (BUHI), Canopy Urban Heat Island (CUHI) and Surface Urban Heat Islands (SUHI), that correspond with the boundary layer they can be monitored/measured in. todo: more

2.1.2 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.

Todo: more

2.1.3 Surface Urban Heat Island (SUHI)

The surface temperature is measured directly at the surface of an object and is the main indicator of the surface urban heat island (SUHI). Surface temperature, in contrast to air temperature, typically is measured via remote sensing technologies like LST via satellites. Well-known satellites include MODIS, Landsat ... (todo list all), that all carry different types of instruments and sensors, that are able to take various measurements. Through the use of satellites, the spatial coverage is great, but raster sizes usually range from one kilometer to a hundred, so the spatial resolution is not that great compared to denser sensor networks. Additionally, these satellites are not geostionary to cover a wide range and therefore only take measurements during a handfull of times a day (todo: number of fly overs with example). Another downside is that satellites in many cases need clear-sky condidtions to measure surface temperature at ground level, as their sensors are not able to penetrate the cloud surface. As a solution, some technologies such as LIDAR

(todo check if true) offer the approximation of the underlying surface temperature below clouds by measuring the out-going radiation of the surface. todo: List of measurement types (microwaves etc.) with disadvantages
Surface and air temperatures can vary greatly, therefore a SUHI does not necessarily also imply a CUHI. Especially in extreme heat events, LST and air temperature can deviate greatly [Goo16].

2.2 Smart Cities

2.2.1 Architecture Layers

Sensing Layer

Data Transportation Layer

Data Management Layer

Application Layer

2.2.2 Applications for ML-based Interpolation

UHI Detection in the Context of Smart Cities

Smart Cities are 'urban areas that exploit operational data [...] to optimze the operation of city services' [HEH+10], by collecting near-real-time data from physical and virtual sensors, integrating those data sources into an enterprise computing platform and performing complex analytics on them. With current urbanization trends (60% ofd ppl living in cities 2060) and the ongoing global and urban warming (cite), research into smart cities has gotten a lot of attention recently (cite). On of the driving factors behind smart cities is next to progress in digitisation of cities the availibility of cheap smart sensors (cite), that enable a good spatiotemporal surveillance of factors in a smart city.
- goals and pillars - architecture - challenges
  - classification of UHI detection into the smart city framework
  In the context of UHI detection,
  - testbeds, sensor networks (citizen owned)
  - cross over to pollution detection

2.3 Interpolation of Missing Data

In the context of urban environments, there are many measurements such as surface temperature, which are measured by sensors that have certain weaknesses. In the case of LST data collection, clouds play a mayor part and prevent surface temperature to be measured. In many LST data-sets (ref), measurements for cloud areas are simply defined as having no value. As many applications and algorithms need continous input data to

work as expected, in this work we take a look on how interpolation, especially with the help of ML, can help solve this problem. In the context of LST data, this could mean interpolating the missing data either based on surrounding data (cite) or by integrating many different features such as air temperature, humidity, heat flux etc. in a ML model (cite).
- moving sensors for better spatial coverage into unobserved areas

Interpolation vs. Extrapolation

In this work, we focus on interpolation of missing data. Interpolation is the process of calculating/guessing missing values between given values. This could be a missing value in a time-series or a missing cell in a data grid (cite). In contrast, extrapolation is the prediction of values based on previous values, like predicting values based on historic time-series data, es in the case of weather forecasting. Due to time constraints, we only focus on the interpolation part, tho extrapolation could also play an important role in smart cities by predicting UHIs and warning citizens about potential future heat-related risks.

2.3.1 Regression Analysis in Statistics

Fundamentaly, regression analysis has its roots in mathematics, more specifically in the field of statistics. todo: more
- foundation of other research fields, based in statistics/mathematics
- linear regression (least-squares) - multiple regression models - hierachical regression
- special cases - piecewise linear regression - inverse prediction - weighted least squares
- logistic regression - poisson regression

2.3.2 Interpolation in Geostatistics

- spatiotemporal (kriging) - time series prediction vs interpolation of missing data - based on GIS - pipeline: fit measured data points to grid, interpolate missing squares downside: cannot find local maximas/minimas

2.3.3 Interpolation with Machine Learning Models

- ML regression

3 System Architecture

The main goal of this work is to apply ML-based interpolation techniques to the topic of UHI detection to improve data availibility and compare different interpolation techniques that are based on traditonal (statistical) and ML-based approaches. ML-based models first of all need a lot of data to be trained and validated with, and after deployment need access to relevent real-time data, if near real-time capabilities are desired. In the context of smart city, such ML-based interpolation models could be used to improve data availibility by interpolating missing/unavailable data such as LST readings under cloudy conditions, and be incoperated into a service that other services, like a UHI detection service, could further rely on, without the need to implement interpolation techniques themselves. This could reduce costs to develop such depending services, as they no longer need to deal with missing data themselves while also improving interpolation results with well-trained and designed models. First, we need to take a look at how such a service fits into existing smart city architectures.

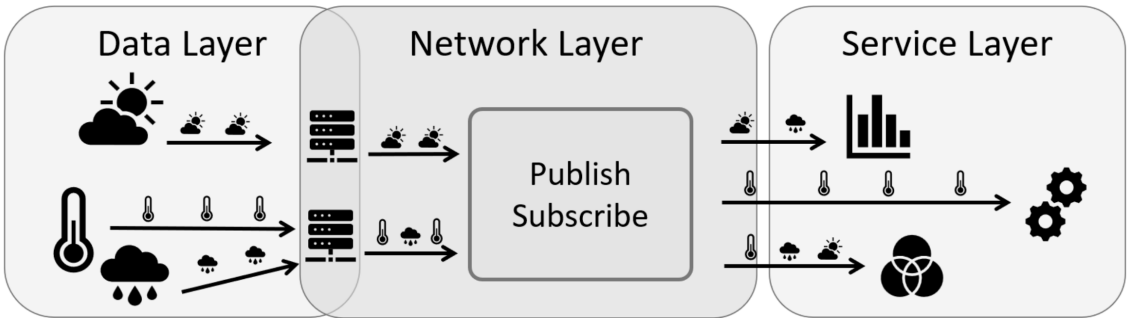

Figure 3.1: In the data layer (left), a wide variety of environmental data is collected with the help of multiple sensors. These are connected to their citizen-owned local base stations, which manage access rights and forward collected data to subscribed services (right) via the decentralized publish-subscribe in the network layer (center).

3.1 Architecture Layers of Smart Cities

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... Figure 3.1 shows the architecture of the SANE system.

3.1.1 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) have seen a lot of attention for many different applications such as 'military sensing, physical security, air traffic control, traffic surveillance, video surveillance, industrial and manufacturing automation, distributed robotics, environment monitoring, and building and structures monitoring' [CK03]. The challenges for WSNs primarily depend among other things on the deployment. An ad-hoc WSN has energy and bandwidth 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 [CMY+15].
In recent years, low-cost sensors (LCS) in combination with sensor networks have enabled fine-granular real-time monitoring of urban environments, although the quality of individual low-cost sensors can be questionable [CDS+17]. In general, LCSs can improve data availibility and support analysis, but do not substitute well-calibrated reference instrumentation [LPS18].

Stationary Sensor Types

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 empirical 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 altitude, the type of sensor used to take the measurement, and the sampling rate. 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 compared to callibrated reference-grade high cost sensors, 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.

- discussion premium vs low-cost sensors - discussion static data incorporation -> NDVI from geoportal via data management layer

- P2,5: highly depends on air flows - humidity: needs to be replaced more often due to pollution - temperature

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.

Modelling of the Sensing Layer

A model is an abstraction of the real world that helps us analyse and understand complex problems. In the context of UHI detection, we can use model climate on a bigger scale, or the micro-slimate inside a city, to better understand and capture influences. In this work, we focus on ...
One major consideration in this context, what type of data is used to validate and train the ML models. The options in this case are either climate models which simulate data, or capturing/collecting real-world data. Due to the complexity and limitations of simulation, this work only focuses on collecting real-world data.

3.1.2 Data Transportation Layer

Todo (small)

3.1.3 Data Management Layer

How to handle such large amounts of data Connection to other APIs Dealing with the heterogenity of data modeling uncertainty of data

Integrating Static Data

Next to sensor readings captured for a given area, there are also other types of information that could be useful for ML applications. Alonso and Renard [AR20] used these

additional indexes to predict air temperature:

- Vegetation Index
  - Normalized Difference Vegetation Index (NDVI)
  - Soil Adjusted Vegetation Index (SAVI)
  - Enhanced Vegetation Index (EVI)
  - Tasseled Cap Transformation greenness (GVI)
  - Density of low vegetation
  - Density of medium vegetation
  - Density of high vegetation
- Water Presence Index
  - Modified Normalized Difference Water Index (MNDWI)
  - Normalized Difference Water Index (NDWI)
- Moisture Index
  - Tasseled cap Transformation Wetness
  - Normalized Difference Moisture Index (NDMI)
- Bare Soil Index
  - Normalized Difference Bareness Index (NDBaI)
  - Bare Soil Index (BI)
  - Enahnced Build-Up and Bareness Index (EBBI)
  - Density of bare soil
- Radiation Index
  - Spectral radiance
  - Emissivity
  - Tasseled Cap Transformation Brightness
- Building Index
  - Normalized Difference Build-Up Index (NDBI)
  - Urban Index (UI)
  - Index-based Build-Up Index (IBI)
  - Building Density
- Topographic
  - Slope (°)

  - Exposure
  - Curvature
- Urban morphology
  - Sky View Factor
  - Standard Deviation (STD) of Building Height (building height variation)
- Land use
  - Distance to railway tracks
  - Distance to points of tourist interest
  - Distance to subway entrances
  - Distances to fountains
  - Water area

These types of data can be sourced either directly via satellites like LiDAR or Landsat 8, or via geoportals that publish such data, like the State Office for Geoinformation and Surveying Hamburg [1] on a regional basis or the EU Inspire Geoportal [2] on a continental basis.

Integrating External APIs

Currently, there exist some cities that already operate open-source sensor networks that usually act as a middleware to upload and retrieve sensor data and offer additonal functionalities like alert or notifications. A detailed list of case studies is provided in 3.3. Next to city specific initiatives, there exist also independent sensor network platforms or initiatives. These are either operated by vendors of f.e. IOT devices, or by open-source communities. In the following, one vendor and one open-source community project is portraied in more detail:

Netatmo: Netatmo [3] is a French Smart Home Company that was founded in 2011. It has a large smart home assortment including cameras, door bells, smoke detectors, thermostats and weather stations among others. In the context of collecting meteological data, the smart weather products are of particular interest. These include a smart 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 3.1.
Complementary to its smart products, Netatmo also operates a weather platform and

[1] https://geoportal-hamburg.de/geo-online/
[2] https://inspire-geoportal.ec.europa.eu/
[3] https://www.netatmo.com/

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 3.1: Netatmo Sensor Specifications (Vendor reported)

a developer portal. The weather platform offers a weather map [4] containing the measurements of connected weather stations accross the whole world for air temperature, precipitation, and wind speed and directon. The developer portal [5] offers a way to programmatically access all sensor measurements via a REST API. In this work, we later use this REST API to collect sensor data, as discussed in Chapter 5.

Next to commercial vendors, there also exist OpenSource projects which operate and develop sensor platforms independently:

Sensor Community: Sensor.Community [6] is an open-source community driven project that aims to collect Open Environmental Data. The project is part of the initiative OK Lab Stuttgart [7], which is run by a group of volunteers that ... 13.500 sensors world-wide, open source
- BME280 - SHS
Most of the solutions have in common, that ... - accounts, request quotas, historical data?

But, they also differ at... - no common REST standard, no shared accounts, sensor descriptions (placement)

3.1.4 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

[4] https://weathermap.netatmo.com/
[5] https://dev.netatmo.com/apidocumentation
[6] https://sensor.community
[7] https://codefor.de/stuttgart/

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 the topic 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 allow UHI detection algorithms to work, in this work we primarily focus on creating and evaluating an air 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.

3.2 Applications for Machine-Learning

The data management layer publishes historical data and real-time data streams for application service to ingest. Generally, ML can be used for the following tasks in different variations:

- Classification

In this context,
This section is about where ML could be used (Land cover,)
- comparison to NLP -> pipeline vs singular model (NLP pipeline vs ChatGPT)
applications: - all-in one solution (all features included) -> huge DL model with billions of parameters (ChaptGPT) - not feasible in this thesis - pipeline solution (use ML models incrementally and build upon results) - similar to NLP - feasible for specific topics - bottom-up approach (use ML to increase the amount of available data)
- types of ML utilisation - utlier detection - classification - regression - interpolation of missing data - prediction of future data

3.2.1 Parallels to Natural Language Processing

NLP pipeline (stemmer, lexer, etc.) vs single model (ChatGPT) - neural network with billions of parameters

3.3 Smart City Case Studies

Many european cities, like Amsterdam, Barcelona, London and Stockholm, have developed their own smart city strategies and platforms. In the following, we take a closer look at the steps each city has taken:

Barcelona, Spain: Barcelona operates the 'Sentilo' platform [8], an open-source smart city platform that aims to break down informatioal silos inside the city to provide unified access to the data available. It currently contains more than 27.000 sensors [9]. It offers a modular and extensible architecture, high performance and scalability, alerts, stats, triggers, a simple REST API to send and receive sensor data, and is cross-platform by being build with Java, Redis and MongoDB.

Amsterdam, Netherlands: Amsterdam utilizes sensors to run an extensive smart lighting system, that

London, England: More infos

Stockholm, Sweden: More infos

More honorable mentions from outside the EU:

Singapore: more infos

San Fransico, US: more infos

[8] `http://www.sentilo.io/`
[9] `http://connecta.bcn.cat/connecta-catalog-web/stats/` (*Accessed: 23.03.2023*)

4 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 model will be compared against 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.

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, which imitate the structure of the human brain, to learn from data and perform various tasks.

4.1 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 the single data points from ground-based weather stations into continous layers. Interpolation can also be applied to singular 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 to simulate individual virtual sensors that are derived from either low-temporal resolution stationary sensors or moving sensors to increase the temporal resolution of individual sensors and the spatial coverage of sensor networks

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 set a focus on the ability to create virtual sensors from moving sensors in order to increase spatial coverage. 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. Afterwards, we will decide on one model that will be implemented and evaluated in Chapter 6.

4.2 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?:** Do we want to model each sensor individually, do we want to model all sensors in the area or do we want to model the whole area at once?

- **How to model stationary vs. non-stationary sensors?:** Do we model stationary and non-stationary sensors differently and can the model handle the possibly unsteadily data of moving sensors, e.g. different locations or different time intervals?
- **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 a 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 f.e. the amount of percipitation influences the humidity.
- **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 4.4.6. 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.
- **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.

- **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

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

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 (cite), like it is the case for temperature interpolation. Generally speaking, this problem can be classified as a supervised learning problem, therefore the possible algorithm candiates are as follows:

- Linear Regression
- KNN Regression
- Neural Networks
  - LSTM
  - RNN

Additonally, there are also other regression such as Regression Trees and Random Forests, Tree Boosting, or Support Vector Regression, which are less popular and seem to be less suitable for the interpolation problem at hand. 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. These models will then later be implemented and compared in chapter 6.
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 for topics such as intelligent transportation systems, smart grids, smart city health care and more can be found in [UATMG20], which unfortunately does not cover interpolation.

4.3 Comparison of Machine Learning Algorithms

4.3.1 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:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \epsilon \tag{4.1}$$

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 4.4.4.
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].

4.3.2 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:

$$\hat{y} = \frac{1}{k} \sum_{i=1}^{k} y_i \tag{4.2}$$

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, however the standard model does not have weights, e.g. all nearest neighbours have the same influence on the prediction, as well as each feature is weighted equally. In the context of correlated input features, this could introduce a bias into the model, if not handled properly in the feature engineering process.

4.3.3 Regression Trees and Random Forests

Decision trees are used for a wide variety of classification problems. Due to the fact that decision trees can vary significantly given similar inputs, e.g have a high variance, random forests were introduced as a counter measure. Random forests combine multiple decision trees and training them on different features and subsets of data and average their predictions in order to reduce the variance (cite).
When instead of a discrete target variable a continous target variable is predicted, regression trees can be used. The principle behind regression trees is to split the data into continously smaller sub-sets, and asking the correct questions at the right time in order

to find the best split. 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:

$$\hat{y} = \sum_{m=1}^{M} c_m \mathbb{1}(x \in R_m) \tag{4.3}$$

Regression trees are easy to understand and interpret. Additionally, they do not require any scaling of variables, which is a big advantage compared to other models.
The mayor disadvantages of the model are for one, that due to the averaging of different regression trees, the model cannot be used to extrapolate data, e.g. predict temperatures outside of the training data. This means that the model needs to be trained with a wide variety of data that includes the maximum and minimum extreme values for the given target variable. As these values might not be easily available, as extreme weather events might occur very infrequently, this could be a problem. Another disadvantage is that regression trees have depth as a hyperparameter that needs to be tuned. If the depth is too big, the tree might overfit the data and if the depth is too small, the underlying relationship might not be captured accurately.
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. However, they do not discuss the downsides of regression trees to extrapolate data.

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.

These studies show that regression trees can be used to predict air temperature with a high accuracy on a daily and annual basis for minimum and maximum air temperature with a high spatial resolution, but do not explore the ability to predict actual air temperature for low time resolutions, e.g. hourly or sub-hourly.
Due to the downside of the inability to extrapolate data which is especially critical in the detection of CUHIs, e.g. air temperature maximas, we do not use regression trees in this work. However, other findings such as station density and features used can be used in this work.

There is a systematic bias to underestimate high temperatures and overestimate low

temperatures [ZKBK21, ZL12] Random Forests (RF) [Bre01]

4.3.4 Tree Boosting

4.3.5 Support Vector Machines

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.

4.3.6 Neural Networks

- trade off between interpretability and model capabilities - more data needed to train the network, because there are more weights to train

Neural networks

Deep Neural Networks

- approximation theorem - special form of neural networks with hidden layers
- overparametrization

4.4 Feature Engineering

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 4.3.1 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 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.

4.4.1 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.

4.4.2 Temporal Autocorrelation

4.4.3 Temporal Cross Correlation

4.4.4 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]

4.4.5 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

Neural Network Models

input layer -> hidden layers -> output

The advantages of Multi-layer Perceptron are: - Capability to learn non-linear models. - Capability to learn models in real-time (on-line learning) using partial_fit.

The disadvantages of Multi-layer Perceptron (MLP) include: - MLP with hidden layers have a non-convex loss function where there exists more than one local minimum. Therefore different random weight initializations can lead to different validation accuracy. - MLP requires tuning a number of hyperparameters such as the number of hidden neurons, layers, and iterations. - MLP is sensitive to feature scaling.

- Loss functions/optimzers: Stochastic Gradient Descend, Adam, L-BFGS

- random forest regression ...

4.4.6 Evaluation Metrics

- Mean absolute error, mean absolute relative, mean quared error, r-squared, root mean squared error (RMSE)

4.5 Model Selection

Question: do we want to model each sensor individually, do we want to model all sensors at once or do we want to model the whole area? Do we model stationary and non-stationary sensors differently?

Options: - linear regression - needs independent variables -> PCA - KNN regression - needs independent variables -> PCA - Neural Network - try without independent variables - LSTM -> temporal correlation - CNN -> spatial correlation, temperature map

After listing the selection criteria, the next step is to discuss the possible model candidates and evaluate them based on the selection criteria. The following models are considered:

- linear regression - knn regression - neural networks - deep learning - sequential model - recurrent neural network (RNN) - long short-term memory (LSTM) - convolutional neural network (CNN)

4.6 Machine Learning in Geostatistics

Idea/Hyphothesis: ML outperforms traditional geostatistical models (in certain scenarios) as it is able to capture more complex interdependencies between features and isn't necessarily bound to the (mathematical) assumptions of geostatistical models.

On important point to mention, is that different meteological features have different interpolation techniques, as they beve different (physical) properties. For example, temperature is a scalar value, while wind speed and direction are vector values. Relative humidity on the other hand is a relativ value that is bound between 0 and 1. For percipitation it gets even more complex, as rainfall is highly connected to cloud coverage and movements. In the following, we will take a look at commonly used existing interpolation techniques for different meteological features and discuss data preprocessing steps.

- Dealing with sparse data - Dealing with extreme weather events (e.g. heat waves, blizzards, etc.) - how can a ML model be trained if no such data exists for a given area? -> transfer learning

4.6.1 Temperature Interpolation

- difference between air and surface temperature

surface temperature -> solar radiation, surface roughness, emissivity, soil moisture, soil temperature, vegetation cover, snow cover, and surface slope

air temperature -> surface temperature

4.6.2 Wind Speed and Direction Interpolation

prob not easily archivable with ML -> depends on high/low pressure areas

4.6.3 Relative Humidity Interpolation

Data preprocessing: scale to [0, 1] does not seem to work well with kriging, maybe just nearest neighbor approach/linear, or the input values are just not accurate

4.6.4 Percipitation Interpolation

DeepLearning Lecture - sequence modeling design criteria - handle variable-length sequences - track long term dependencies - maintain information about order - share parameters across the sequence

The model is implemented following advice from [RABW23]...

Additonal ideas: - types of geological features (vectors, rasters) -> distance, containment, intersection, etc.

4.7 Additional Considerations

ML Model Deployment

Deployed as a service. Input ingested -> continous data map as output all 5 min or so (could also be smaller depending on use-case -> trade-off between cost and accurancy)

ML Model Retraing

Need to retrain model from time to time, e.g. if the accurracy drops below a certain threshold.

5 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 of the underlying process that the model should be fitted to, there can be bias or an inability to generalize well to new data. In this chapter, we take a look at potential data sources 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, e.g. annotated datasets as provided by Google Research [1]. In the field of climate research, there are also many datasets, including satellite data, weather station data, and climate model data. 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, as discussed in 3.1.1, 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 seen in figure 5.6 which shows the weather station network of the official german weather services (DWD). However they are met by dense sensor networks. A mayor challenge of running such a dense sensor network in a controlled scientific environment is the potential high cost of individual sensors, e.g. high precision and calibration, and the maintenance of the network, e.g. battery replacement, damages due to environmental influences, theft etc. Therefore, there are very few such projects and currently no openly available datasets. There were projects in the past such as the Helsinki Testbed [KPS+11] or the Birmingham Urban Climate Laboratory [WYC+16], but links to their datasets are unfortunately not maintained, showing the difficulty of finding and retrieving relevant datasets in this field.
As an alternative, in this work we instead mainly use data from citizen run sensor networks, e.g. crowdsourced data from personal weather stations (PWS), which are either openly available or can be requested in limited fashion via provided APIs. In this work we create train and test datasets by combining openly available sensor data from the providers Sensor Community, Netatmo, and the official german weather services (DWD) for quality control. Other sources for crowdsourced weather station data include Weath-

[1] https://research.google/resources/datasets/

erObservationsWebsite (WOW) [2] and Weather Underground [3]. 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 5.1. 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.

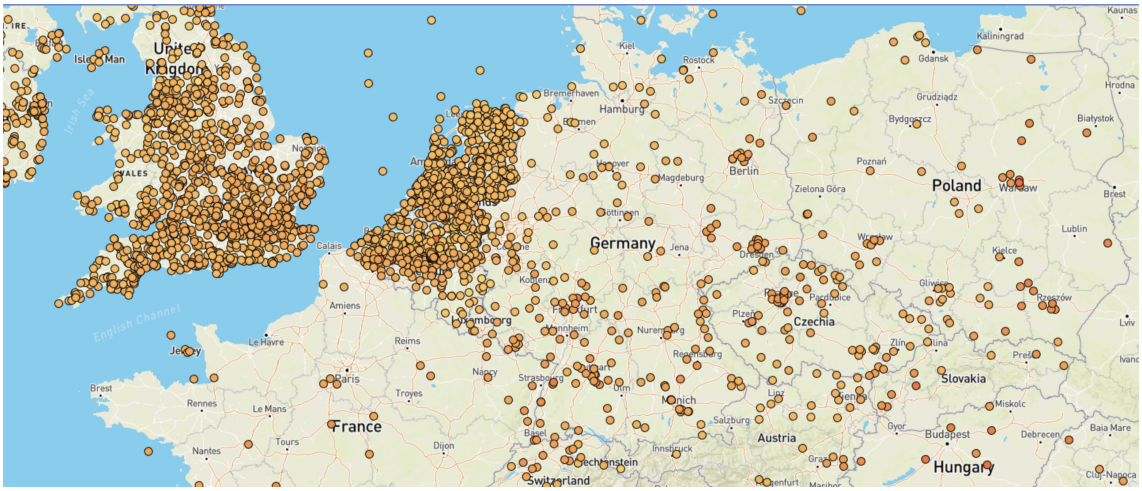

Figure 5.1: Temperature sensor locations from WOW, acessed on 05.07.2023

5.1 Sensor Community

Sensor Community [4] is a contributers driven global sensor community that creates Open Environmental Data, and has an archive available [5] of their historical sensor data world wide. There are no quality measures recorded for each sensor, but as crowd-sourced sensor data tends to have a lower quality than professsionally setup sensors, e.g. sensor placement by non-professionals, we need to explore how the data quality looks like. In Figure 5.2, 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 fig. 5.3, 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

[2] https://wow.metoffice.gov.uk/
[3] https://www.wunderground.com/
[4] https://sensor.community/en/
[5] https://archive.sensor.community/

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 %“.

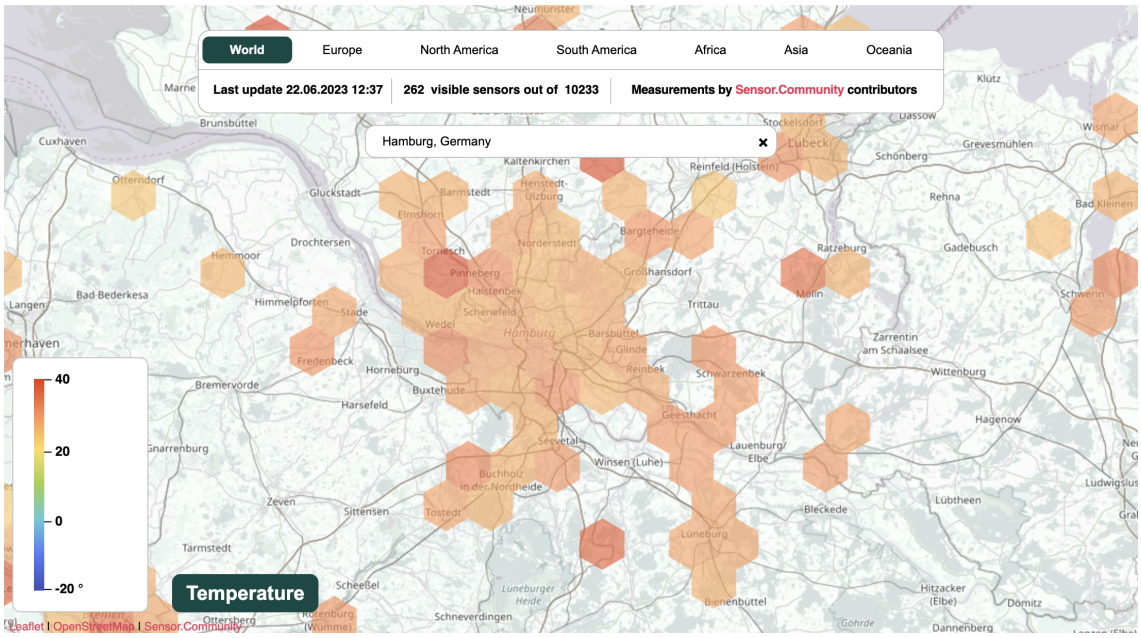

Figure 5.2: Temperature map from Sensor Community for Hamburg, Germany, on 22.06.2023 12:51h with the DWD reference at 25°C

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

Figure 5.3: 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 [6]. Of these sensors, many are located in Germany and almost half of them are of type BME 280, which is a lost-cost Bosch sensor which can measure temperature, pressure and humidity, as seen in appendix 1. The sensor locations as of may 2023 are shown in 5.4. 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.

[6] as of 24.06.2023

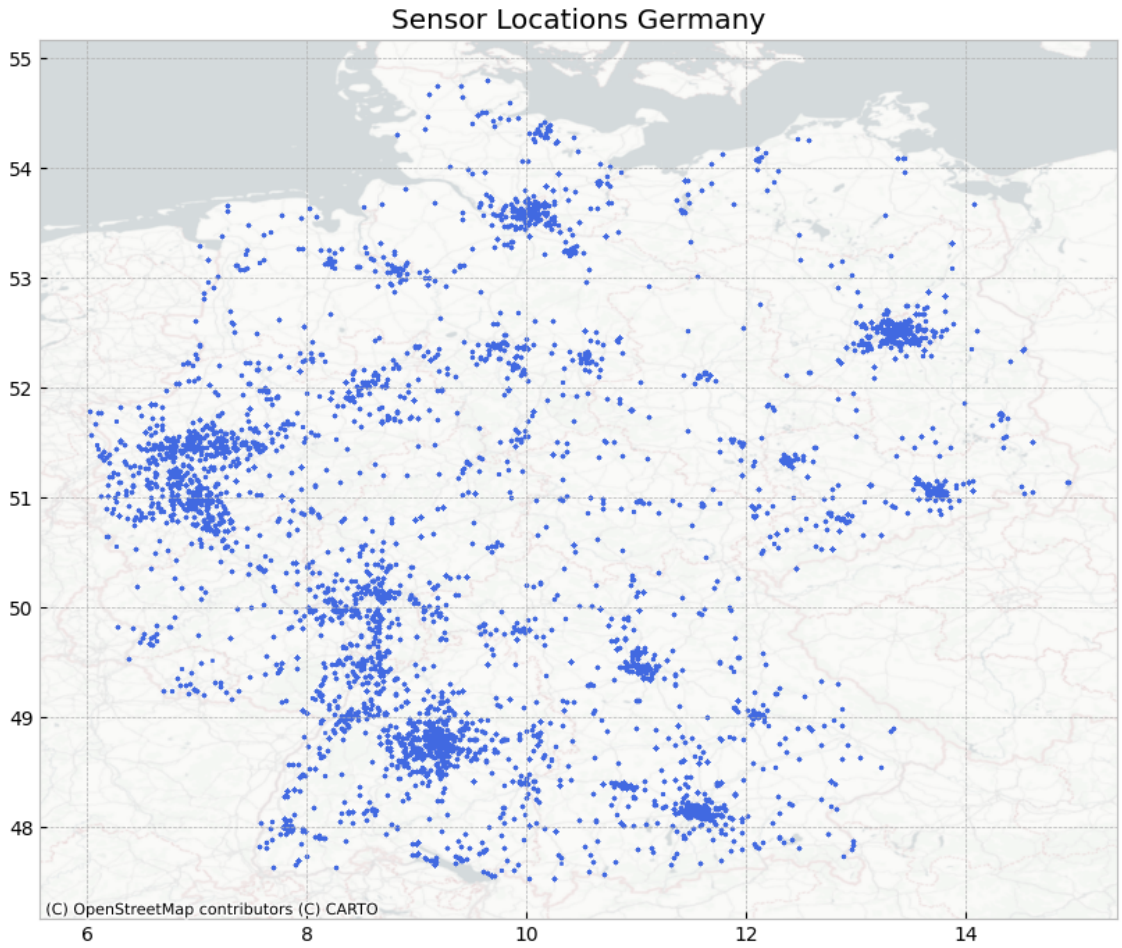

Figure 5.4: Sensor locations of Sensor Community in Germany, as of 01.05.2023, of sensor type DHT22 (2590 sensors), BME280 (1558 sensors), BMP280 (100 sensors), BMP180 (72 sensors)

5.2 Netatmo

Netatmo [7] is an american company that sells smart-home devices including weather stations. They host a weather map [8] where customers can share their weather station data. The provide an API to access current weather station data. They provide historical and current weather data for commercial partners or partners in the research and education sector. They are part of the EUMETNET project [9] which is a network of 31 European meteological 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, only private datasets such as `https://catalogue.ceda.ac.uk/uuid/e8793d74a651426692faa100e3b2acd3` (last acessed 05.07.2023), that are only

[7] `https://www.netatmo.com/en-eu`
[8] `https://weathermap.netatmo.com/`
[9] `https://www.eumetnet.eu/`

available for EUMETNET members. They offer an educational program [10] to acess limited amounts of data that is usually only available to commercial partners.

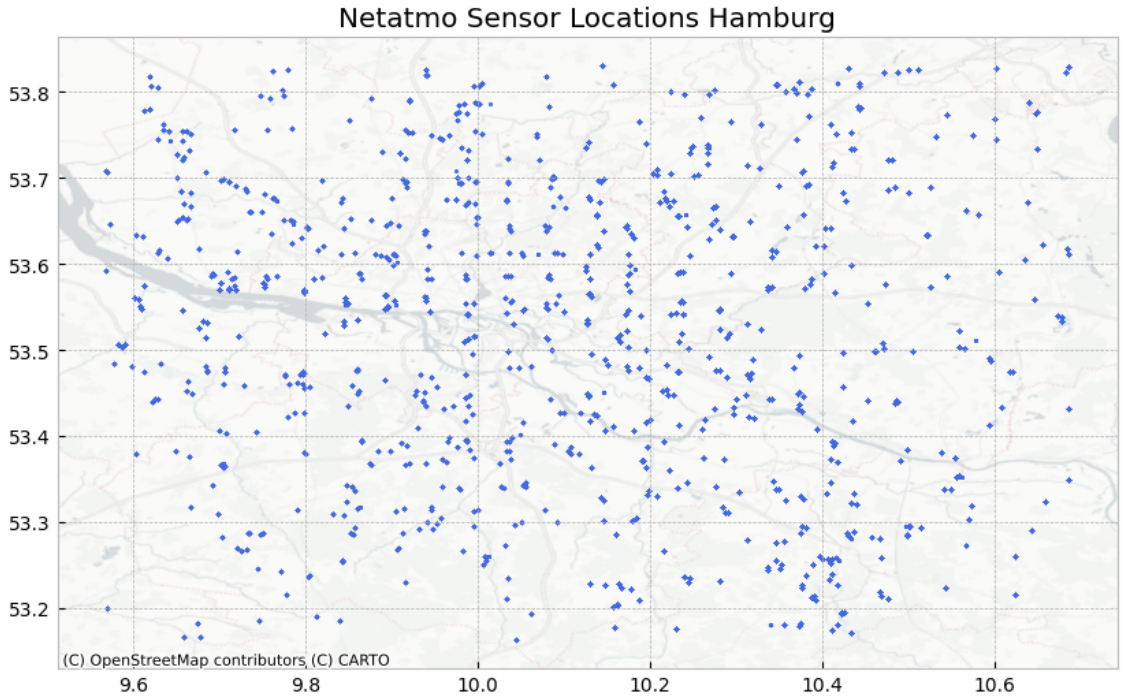

Figure 5.5: Sensor locations of Netatmo in Hamburg, Germany, as of 28.06.2023

In this work, data from Netatmo stations is primarily used as Netatmo offers a large amount of sensors in Germany, as seen in figure 5.5. Due to the limitations of the API rate limits, only data for the region of Hamburg, Germany is used.

5.3 DWD

- german official weather service - high data quality - low spatial coverage

5.4 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

[10] https://www.netatmo.com/en-eu/weather-with-netatmo

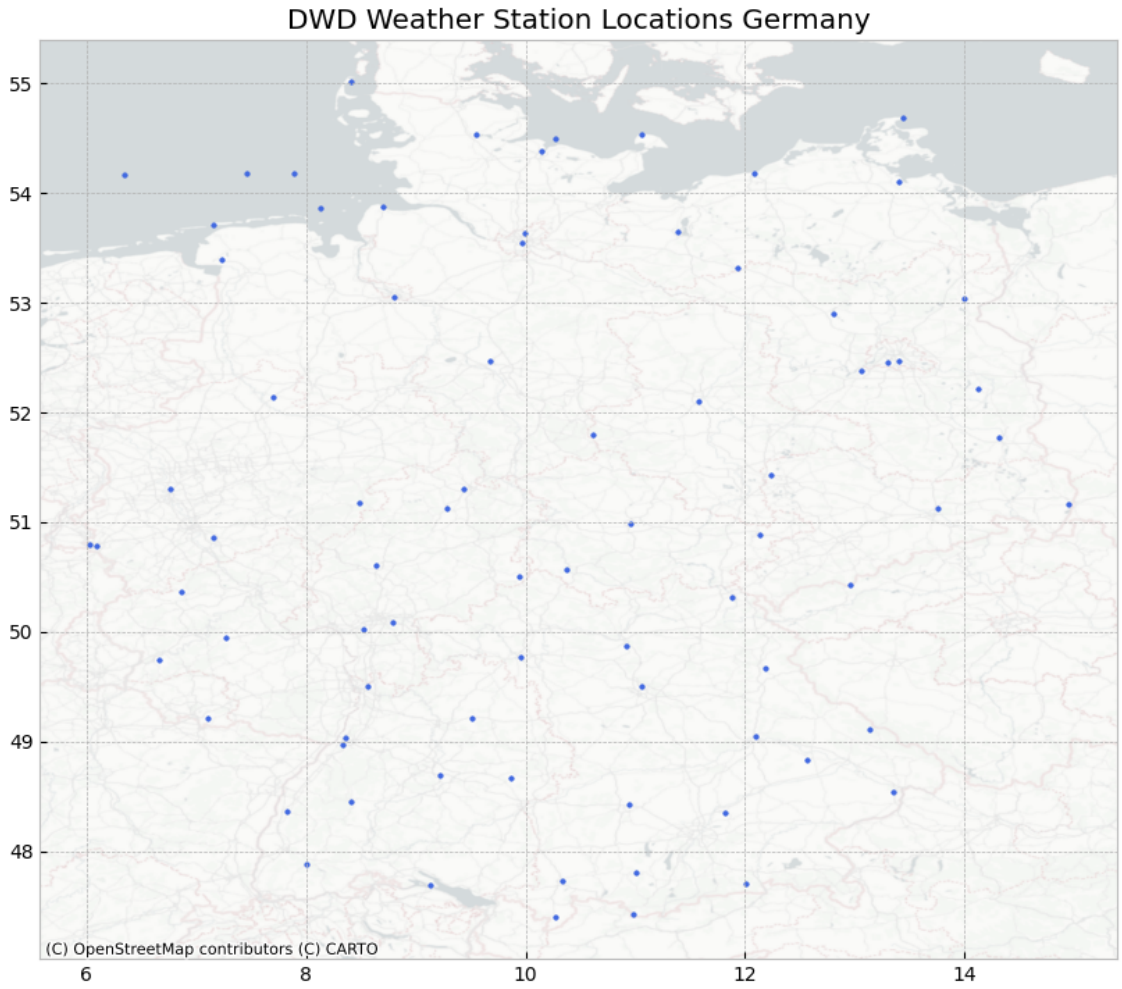

Figure 5.6: DWD Weather Station Locations in Germany, `https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/subdaily/standard_format/KL_Standardformate_Beschreibung_Stationen.txt`, accessed 28.06.2023

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 data sources [HGMS+22]. The following tools are available:

- CrowdQC (R package [11])

[11] `https://doi.org/10.14279/depositonce-6740.3`

- CrowdQC+ [FBD+21] (R package [12])
- TITAN (R package [13])
- NetatmoQC (Python 3 package [14])

CrowdQC was used in ...
The usage of QC lead to different percentages of data being flagged as outliers

5.5 Overview Data Sources

With the rise of the Internet and Big Data, the amount of available data has increased exponentially in the last decade. Additionally, movements such as the OpenData movement have made many datasets publicly available and promoted collaboration in research. There are many different data sources available that can be categorized as fully open source, OpenData, or commercial.
Open source data sets can be found via... A very popular example is the OpenStreetMap project, which aims to ...
OpenData...
Example for commercial data providers are Netatmo...

Reference data: DWD station data (see https://dwd-geoportal.de/products/OBS_DEU_PT10M_T2M/)
-> 10 min 2m height air temperature https://opendata.dwd.de/climate_environment/CDC/observations_ge
need to write script to download all files and for each file do some conversion to csv
- overview of sources for good datasets for temperature and climate related research
goal: - collect mutiple datasets (many features, fine-granualar spaciotemporal) - enhance datasets with additonal information (soil conditions, zoning plans, vegetation health)

5.6 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. 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. The target feature in this work is the air temperature at 2m height. The input features are a combination of sensor readings from weather stations, sensor networks such as Sensor Community and Netatmo, as well as additonal meta data such as soil conditions, zoning plans, vegetation health, and satellite data.

[12] https://github.com/dafenner/CrowdQCplus
[13] https://github.com/metno/TITAN
[14] https://source.coderefinery.org/iOBS/wp2/task-2-3/netatmoqc

- target variable: air temperature - input features: - weather station measurements (air temperature, humidity, wind speed, wind direction, percipitation) - satellite data (surface temperature, surface roughness, soil temperature, land coverage indexes, sky view factor, ...)

Need to separate between reference grade data (weather station calibrated), and low-cost sensors without placement information

6 Evaluation

TODO 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

6.1 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 uses a covariance function to model the spatial correlation between data points. 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, 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
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 [1]:

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

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.

6.2 Model Evaluation

TODO

[1] https://pro.arcgis.com/en/pro-app/latest/tool-reference/geostatistical-analyst/an-overview-of-the-geostatistical-analyst-toolbox.htm

6.2.1 Model Validity

Use 70% of data for training, 30% for test. cross-validation

6.3 Variable Importance

6.4 Uncertainty Analysis

measurement uncertainty, contextual uncertainty, prediction uncertainty

6.5

7 Conclusion

7.1 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.

7.2 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

[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

[Bre01] BREIMAN, Leo: Random forests. In: *Machine learning* 45 (2001), S. 5–32

[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

[CW11] CAMPBELL, James B. ; WYNNE, Randolph H.: *Introduction to remote sensing*. Guilford Press, 2011

[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

[Gea54] GEARY, Robert C.: The contiguity ratio and statistical mapping. In: *The incorporated statistician* 5 (1954), Nr. 3, S. 115–146

[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

[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)

[HEH+10] HARRISON, Colin ; ECKMAN, Barbara ; HAMILTON, Rick ; HARTSWICK, Perry ; KALAGNANAM, Jayant ; PARASZCZAK, Jurij ; WILLIAMS, Peter: Foundations for smarter cities. In: *IBM Journal of research and development* 54 (2010), Nr. 4, S. 1–16

[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

[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

[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

[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

[LF14] LECLERC, Monique Y. ; FOKEN, Thomas: *Footprints in micrometeorology and ecology*. Bd. 239. Springer, 2014

[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)

[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

[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

[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

[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

[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

[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

[RABW23] REY, Sergio ; ARRIBAS-BEL, Dani ; WOLF, Levi J.: *Geographic data science with python*. CRC Press, 2023

[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

[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

[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

[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

[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

[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

[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

[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

[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

Sensor Community

Listing 1: 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 Studiengang XXX 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 und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

Hamburg, den _____ Unterschrift: _____

Veröffentlichung

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

Hamburg, den _____ Unterschrift: _____