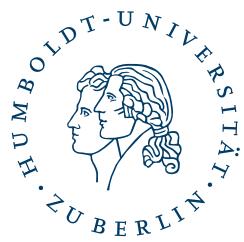


H U M B O L D T - U N I V E R S I T Ä T Z U B E R L I N



Lebenswissenschaftliche Fakultät

Institut für Biologie

Masterarbeit

zum Erwerb des akademischen Grades

Master of Science Biophysics

**DATALAB: ENTWICKLUNG EINER INTEGRIERTEN WEBBASIERTEN
VISUALISIERUNG VON UMWELTDATEN
IM KONTEXT DATA LITERACY**

**DATALAB: DEVELOPMENT OF AN INTEGRATED WEB-BASED VISUALIZATION OF ENVIRONMENTAL DATA
IN THE CONTEXT OF DATA LITERACY**

vorgelegt von

BEUSTER, BENJAMIN

Matrikelnummer 571573

benjamin.beuster@student.hu-berlin.de

geb. am 09.06.1984, Berlin

Erstprüferin: Prof. Dr. Annette Upmeier zu Belzen

Zweitprüfer: Prof. Dr. Burkhard Priemer

Inhaltsverzeichnis

Abbildungsverzeichnis	3
Tabellenverzeichnis	3
Abkürzungsverzeichnis	4
Zusammenfassung	5
1 Einleitung	6
1.1 Relevanz	6
1.2 Entwicklung des Begriffs Data Literacy - ein 21st Century Skill	7
1.3 Einordnung des DataLabs - Abgrenzung zu anderen Tools	8
2 Theoretischer Hintergrund	10
2.1 Probleme im Umgang mit Daten	10
2.2 Didaktische Methoden zur Förderung von Data Literacy und Kriterien zur Visualisierung von Umweltdaten	14
2.3 Zielsetzung & Forschungsfrage	16
3 Material und Methoden	16
3.1 Konzeption von Modulen zur Visualisierung von Umweltdaten für das DataLab	16
3.1.1 Konzeption: Messstellen-Karte	16
3.1.2 Konzeption: Hexagonale Dichtekarte	19
3.1.3 Konzeption: Berliner Wärmeinseleffekt	20
3.1.4 Konzeption: Live vor Ort - Sensordaten am Standort	21
3.1.5 Konzeption: Live im Vergleich - Sensordaten über Standorte hinweg	22
3.1.6 Konzeption: Interaktive Datenexploration	24
3.2 Implementierung des DataLabs	25
3.2.1 Technologie Stack	27
3.2.2 Struktur des DataLabs	30
3.3 Abruf der API	32
3.3.1 Tabelle aller SenseBoxen	33
3.3.2 Zeitfenster mit einem Delta berechnen	36
3.3.3 Messwerte der SenseBoxen abrufen	37
3.3.4 Messwerte in die Datenbank schreiben	38
3.4 Implementierung der DataLab Module	38
3.4.1 Messstellen-Karte	38
3.4.2 Hexagonale Dichtekarte	40
3.4.3 Berliner Wärmeinseleffekt	42

3.4.4	Live vor Ort – Sensordaten am Standort	44
3.4.5	Live im Vergleich – Sensordaten über Standorte hinweg	47
3.4.6	Interaktive Datenexploration	47
4	Ergebnisse	48
4.1	Messstellen-Karte	48
4.2	Hexagonale Dichtekarte	51
4.3	Berliner Wärmeinseleffekt	53
4.4	Live vor Ort – Sensordaten am Standort	54
4.5	Live im Vergleich – Sensordaten über Standorte hinweg	56
4.6	Interaktive Datenexploration	58
5	Diskussion	60
5.1	Module des DataLabs	61
5.1.1	Messstellen-Karte	61
5.1.2	Hexagonale Dichtekarte	62
5.1.3	Berliner Wärmeinseleffekt	63
5.1.4	Live vor Ort – Sensordaten am Standort	64
5.1.5	Live im Vergleich – Sensordaten über Standorte hinweg	66
5.1.6	Interaktive Datenexploration	67
5.2	Technische Realisation	68
5.3	Anwendungsfreundlichkeit im Unterricht	69
6	Schlussfolgerung und Ausblick	69
Literaturverzeichnis		71
Danksagung		75
Anhang		I
Quellcode des DataLabs		I
Data Literacy Competencies Matrix (Ridsdale et al., 2015)		II
Eigenständigkeitserklärung		

Abbildungsverzeichnis

1	Auswahl von Schlüsselqualifikationen, die für das DataLab relevant sind, aus der Data Literacy Kompetenz Matrix nach Ridsdale et al. (2015), den darin enthaltenen Kompetenzen, den daraus abgeleiteten Kenntnissen und Aufgaben, sowie den Problemen, die Lernende im Umgang damit haben.	11
2	Der Softwarestack des DataLabs.	30
3	Darstellung der wesentlichen Ordnerstruktur des DataLabs.	31
4	Auszug einer in JSON formatierten Antwort der openSenseMap-API.	34
5	Messstellen-Karte mit blauen Markierungen für Messstellen.	49
6	Liniendiagramm, wie es im DataLab dargestellt wird.	51
7	Darstellung von Temperaturwerten in Form von Sechsecken auf einer Landkarte in einer Animation über die Zeit.	52
8	Berliner Wärmeinseleffekt um 14 Uhr an einem typischen Sommertag.	54
9	Erfrischungskarte mit Animation (Ausschnitt).	55
10	Darstellung von <i>Live vor Ort – Sensordaten am Standort</i> während einer laufenden Unterrichtseinheit.	57
11	Darstellung von zwei Standorten mit dem Modul <i>Live im Vergleich – Sensordaten über Standorte hinweg</i>	59
12	Auswahlmöglichkeiten zur Erstellung eigener Diagramme.	60

Tabellenverzeichnis

1	Überblick über ausgewählte Tools und Plattformen zur Förderung von Data Literacy im Schulkontext.	9
2	Didaktische Methoden zur Förderung von Data Literacy, abgeleitet aus ausgewählten Studien.	15
3	Kriterien zur Visualisierung von Umweltdaten im Kontext Data Literacy, abgeleitet aus ausgewählten Studien.	17
4	Tools des Technologischen Stacks und dessen jeweilige Beschreibung und Funktion.	29

Abkürzungsverzeichnis

API - Application Programming Interface

CMS - Content Management System

dict - Dictionary

GIF - Graphics Interchange Format

html - Hypertext Markup Language

json - JavaScript Object Notation (auch JSON)

NaN - Not a Number

ORM - Object-relational mapper

POSIX - Portable Operating System Interface

URL - Uniform Resource Locator

VM - Virtual Machine

Zusammenfassung

Das Ziel der vorliegenden Arbeit besteht in der Konzeption und Implementierung des Data-Labs, einer webbasierten Plattform zur Visualisierung von Umweltdaten. Das DataLab soll zur Förderung ausgewählter Komponenten aus dem Bereich Data Literacy genutzt werden, einem 21st Century Skill, der als Grundlage für faktenbasiertes und wissenschaftliches Denken zunehmend an Bedeutung gewinnt. Perspektivisch soll das DataLab im Rahmen der *Humboldt Explorers*, eines Citizen-Science-Schulprojekts, zum Einsatz kommen. Bei der Konzeption wurden Erkenntnisse aus der Theorie und Empirie berücksichtigt, wobei sowohl auf bekannte Probleme beim Erwerb von Data Literacy eingegangen wird, als auch gängige Kriterien zur Visualisierung von Umweltdaten Beachtung finden. Der Schwerpunkt der vorliegenden Arbeit liegt in der technischen Realisation dieses theoriegeleiteten Vorgehens. Insgesamt werden sechs Module entwickelt, welche darauf abzielen Kompetenzen der Datenerfassung, Datenanalyse, Dateninterpretation sowie den kritischen Umgang mit Datenquellen und Datenqualität zu fördern. Die technische Umsetzung der Konzeption wird detailliert beschrieben, so dass die Implementierung transparent nachvollzogen werden kann. Die realisierten Module werden beschrieben und anschließend in Hinblick auf die Erreichung der angestrebten Zielsetzung diskutiert. Die technische Realisation und Anwendungsfreundlichkeit wird kritisch reflektiert. Abschließend erfolgt die Einordnung der Arbeit in den weiterführenden Forschungsprozess.

The aim of this thesis is to design and implement the DataLab, a web-based platform for the visualization of environmental data. The DataLab is to be used to promote selected components from the field of Data Literacy, a 21st century skill that is becoming increasingly important as a basis for fact-based and scientific thinking. The DataLab is planned to be used as part of the *Humboldt Explorers*, a citizen science school project. The conceptual design considers theoretical and empirical findings, addressing known problems in the acquisition of data literacy as well as common criteria for the visualization of environmental data. The focus of the present work lies in the technical realisation of this theory-based approach. A total of six modules are developed, which aim to promote skills in data collection, data analysis, data interpretation and the critical handling of data sources and data quality. The technical implementation of the concept is described in detail so that the implementation can be easily understood and reproduced. The modules are described and then discussed with the aim of achieving the desired objectives. The technical realization and user-friendliness are critically reflected upon. Finally, the further research process that follows on from the present work is outlined.

1 Einleitung

1.1 Relevanz

In einer Gesellschaft die zunehmend Entscheidungen auf Grundlage von Daten trifft werden die Fähigkeiten Daten zu verstehen, kritisch zu hinterfragen und Handlungen daraus abzuleiten immer wichtiger. Diese Fähigkeiten werden unter dem Begriff Data Literacy zusammengefasst und sind zu einer Schlüsselkompetenz für Bildung, Beruf und gesellschaftliche Teilhabe geworden.

Verschiedene Studien und bildungspolitische Strategien heben vor diesem Hintergrund die Relevanz dieser Fähigkeiten im schulischen Kontext und darüber hinaus hervor.

So stellt etwa die (in regelmäßigen Abständen in Deutschland durchgeführte) *Trends in International Mathematics and Science Study (TIMSS)* (Schwippert et al., 2024) die Wichtigkeit von datenbezogenen Kompetenzen dar. Der „Umgang mit Daten“ zählt zu den drei zentralen Bereichen des Mathematikunterrichts in der Grundschule. Die Studie schlussfolgert, dass Aufgaben aus dem Bereich der Data Literacy, wie Anwenden, Interpretieren und Problemlösen mit Daten, schon früh gefördert werden müssen.

Börner et al. (2019) argumentieren, dass die Fähigkeit zur Interpretation und Visualisierung von Daten im digitalen Zeitalter ebenso wichtig ist wie lesen und schreiben. Der sichere und reflektierte Umgang mit Diagrammen und anderen komplexen Visualisierungen ist notwendig, um evidenzbasierte Entscheidungen in Alltag, Beruf und Wissenschaft zu treffen.

Auch die Kultusministerkonferenz beschreibt in ihrer Strategie für „Bildung in der digitalen Welt“ (Kultusministerkonferenz, 2016) klare Anforderungen an den Unterricht. Der Umgang mit digitalen Informationen und Daten ist „zentrale Voraussetzungen für soziale Teilhabe“, die zeitgleich zur Vermittlung von Lesen und Schreiben in vielfältigen Lerngelegenheiten systematisch aufgebaut werden sollten. Der vorgestellte Kompetenzrahmen enthält Elemente wie Daten lesen, bewerten und interpretieren und damit Kernelemente der Data Literacy.

Masnick et al. (2017) zeigen in ihrer Arbeit, dass insbesondere die kritische Interpretation von Daten nicht etwa intuitiv gegeben ist, sie muss gezielt vermittelt und gefördert werden. Sie zeigen, dass die Arbeit mit Daten in einer zielgerichteten und wissenschaftlich geprägten Art und Weise die Fähigkeit zur Überprüfung und zur Korrektur von Überzeugungen föhren kann. Data Literacy wird damit zu einer Grundlage für faktenbasiertes und wissenschaftliches Denken.

Wie die Studie von Garcia-Mila et al. (2016) zeigt, sind das Lesen und Verstehen von Informationen aus Tabellen und Diagrammen gesellschaftlich relevante Kompetenzen. So wurden in ihrer Arbeit Teilnehmende dazu angehalten anhand bereitgestellter Daten Pro

und Contra Argumente für den Bau von Atomkraftwerken zu entwickeln. Die Autor:innen heben hervor, dass Lernende in der PISA-Studie Defizite im Umgang mit Daten aufweisen, was die Notwendigkeit von einer gezielten Förderung unterstreicht.

Die Studien verdeutlichen: Data Literacy ist keine rein fachspezifische Fähigkeit, sondern eine zentrale Schlüsselkompetenz, die kritisches Denken stärkt und gesellschaftliche Teilhabe ermöglicht – ein unverzichtbarer 21st Century Skill (Schüller et al., 2021). Insbesondere angesichts polemisch geführter Diskussionen und gesellschaftlicher Spaltungsphänomene, wie sie exemplarisch in Hinblick auf den Klimawandel zu beobachten sind, erscheint die gezielte Förderung von DataLiteracy bei jungen Menschen als notwendig.

1.2 Entwicklung des Begriffs Data Literacy - ein 21st Century Skill

Eine der ersten systemtischen Definitionen des Begriffs Data Literacy stammt von Mandinach und Gummer (2013). Sie definieren Data Literacy als eine Sammlung von Fähigkeiten (Skills) und einer Wissenbasis (knowledge base), die es ermöglichen Daten in Informationen und schließlich in anwendbares Wissen umzuwandeln. Zu den nötigen Fähigkeiten gehört es Daten zu finden, zu sammeln, zu organisieren, zu analysieren, zusammenzufassen und zu priorisieren. Dazu gehört auch das Wissen, wie Hypothesen gebildet werden, Probleme identifiziert werden können, wie Daten interpretiert, sowie Maßnahmen festgelegt, geplant, durchgeführt und überwacht werden können. Deahl (2014) erweitert diesen Begriff um den Kontext, wie diese Fähigkeiten im Unterricht vermittelt werden können. Nach ihr ist der Begriff definiert, als die Fähigkeit Datenquellen zu finden, Daten zu verstehen, zu sammeln, zu interpretieren, zu visualisieren und die Möglichkeit qualitative und quantitative Daten zur Argumentation zu nutzen. Deahl (2014) führt weiter aus, dass über das einfache „Daten verstehen“ hinaus gegangen werden soll und Daten auf mögliche Verzerrungen und Ungenauigkeiten hin geprüft und Messungen selbst durchgeführt werden sollen. Vahey et al. (2006) sehen in Data Literacy die Fähigkeit, Fragen und Antworten durch die Verwendung von Daten und evidenzbasierter Denkweise zu stellen und zu beantworten, um reale Probleme zu lösen und die Lösungen kommunizieren zu können. Ridsdale et al. (2015) verwenden all diese Ansätze und Definitionen und schlagen eine allgemeingültige Definition vor:

„Data literacy is the ability to collect, manage, evaluate, and apply data, in a critical manner.“ (*Data Literacy ist die Fähigkeit, Daten auf kritische Weise zu sammeln, zu verwalten, zu bewerten und anzuwenden.*)

Diese Definition von Data Literacy soll für die vorliegende Arbeit verwendet werden.

1.3 Einordnung des DataLabs - Abgrenzung zu anderen Tools

Die vorliegende Arbeit widmet sich denjenigen Schlüsselqualifikationen und Kompetenzen aus dem Bereich der Data Literacy, die im Schulunterricht im Projekt *Humboldt Explorers* („Humboldt Explorers“, 2021) vermittelt werden sollen. *Humboldt Explorers* ist ein Citizen-Science-Schulprojekt der Bayer Foundation, der Stiftung Humboldt-Universität und der Humboldt-Universität zu Berlin. Ziel dieses Projektes besteht darin, Lerngruppen zwischen der 5. bis 10. Jahrgangsstufe 21st Century Skills und im Speziellen die darin enthaltende Data Literacy zu vermitteln. Dazu werden Lerneinheiten entworfen, die sich auf die Lebenswelten der Lernenden beziehen und an aktuelle gesellschaftliche Themen anknüpfen. Die Lernenden untersuchen dabei naturwissenschaftliche Phänomene in ihrer Umwelt mit analogen und digitalen Werkzeugen und werden dabei didaktisch angeleitet, um mittels Daten verschiedene Fragestellungen zu bearbeiten und zu eigenen Schlussfolgerungen zu kommen.

Die Tabelle 1 enthält eine Auflistung von bereits bestehenden Tools zur Visualisierung von Daten, die unter Anderem zur Vermittlung von Data Literacy genutzt werden können. Das DataLab, welches im Rahmen der vorliegenden Arbeit entwickelt wurde, grenzt sich von diesen Tools ab, indem es (neben theoriegeleiteten Überlegungen) möglichst spezifisch auf die Anforderungen im Rahmen des Projekts der *Humboldt Explorers* eingeht. So wird im Gegensatz zu den bereits bestehenden Tools ermöglicht, Live-Daten von SenseBoxen einzubinden, die Standorte der Messtationen auf einer Karte einzubinden, Satellitendaten und andere kontextrelevante Informationen mit der Darstellung von lokal aufgezeichneten Sensordaten zu kombinieren und speziell auf den Berliner Wärmeinseleffekt einzugehen.

Tabelle 1: Überblick über ausgewählte Tools und Plattformen zur Förderung von Data Literacy im Schulkontext.

Tool / Plattform	Zielgruppe	Schwerpunkt / Beschreibung	Link
CODAP	Sek I / Sek II	Plattform für explorative Datenanalyse, für Projektarbeit, Zeitleisten, Diagramme	https://codap.concord.org
DataBasic.io	Sek II / Lehrkräfte	Einführung in Daten mit Mini-Tools (etwa Textanalyse, Muster finden)	https://databasic.io
Flourish	Sek II / Projektarbeit	Erstellung eigener Visualisierungen	https://flourish.studio
Gapminder Tools	Sek I / Sek II	globale Datensätze, visualisiert als animierte Zeitverläufe, geeignet für Diskussionen über soziale und globale Themen	https://gapminder.org/tools
Google Data Explorer	Sek II	Zugriff auf offene Datensätze, einfache Visualisierung	https://datacommons.org
RAWGraphs	Sek II	Open-Source-Tool zur Datenvisualisierung, komplexere Darstellungen für Lernende mit Vorkenntnissen.	https://rawgraphs.io
Seeing Theory	Sek II	Einführung in Statistik, theoretischer, aber anschaulich	https://seeing-theory.brown.edu
Tuva Labs	Sek I / Sek II	fertige Unterrichtseinheiten zur Datenanalyse mit leicht verständlicher Oberfläche	https://tuvalabs.com

2 Theoretischer Hintergrund

2.1 Probleme im Umgang mit Daten

Ridsdale et al. (2015) stellen eine Data Literacy Kompetenz Matrix auf. Sie nennen mehrere Schlüsselqualifikationen, denen sie wiederum erforderliche Kompetenzen zuordnen. Zu jeder Kompetenz werden die dafür notwendigen Kenntnisse und Aufgaben erwähnt.

In Abbildung 1 findet sich eine tabellarische Übersicht, die eine für das DataLab relevante Auswahl von Schlüsselqualifikationen aus der Data Literacy Kompetenz Matrix nach Ridsdale et al. (2015) auflistet. Zu jeder der genannten Schlüsselqualifikationen werden die notwendigen Kompetenzen und daraus abgeleiteten erforderlichen Kenntnisse und Aufgaben aufgeführt. Des Weiteren wurden mittels Literaturrecherche Probleme identifiziert, die bei Lernenden im Umgang mit Daten beobachtet wurden. Diese Probleme werden den von Ridsdale et al. (2015) definierten Kenntnissen und Aufgaben zugeordnet. Diese Zusammenstellung des aktuellen Forschungsstandes soll dazu beitragen, im Rahmen der vorliegenden Arbeit möglichst passgenaue Module zu entwickeln, welche die aufgeführten Kenntnisse und Aufgaben vermitteln und dabei die zu erwartenden Probleme berücksichtigen.

Als erste Schlüsselqualifikationen wird in Abbildung 1 das *konzeptionelle Rahmenwerk* aufgeführt. Dieser Qualifikation wird die Kompetenz *Einführung in Daten* zugeordnet. Für diese Kompetenz ist das grundsätzliche Verstehen von Daten, sowie die Kenntnis und das Verständnis über den Gebrauch und die Anwendung von Daten notwendig. Lachmayer et al. (2007) und McDermott et al. (1987) argumentieren, dass Lernende hier Probleme bei dem allgemeinen Verständnis von Diagrammen und inhaltlichem Wissen haben, sowie Probleme mit der Informationsentnahme aus Diagrammen.

Schlüsselqualifikationen	Kompetenzen	Kenntnisse & Aufgaben	Probleme im Umgang mit Daten
Konzeptionelles Rahmenwerk	Einführung in Daten	Kenntnis und Verständnis von Daten Kenntnis und Verständnis über Gebrauch und Anwendung von Daten	Informationsentnahme aus Diagrammen: Achsen, Skalen und Variablen erkennen (Lachmayer et al., 2007) Allgemeines Verständnis von Diagrammen, die Art des verwendeten Diagramms und inhaltliches Wissen über die Darstellung (McDermott, Rosenquist, & van Zee, 1987)
Sammeln von Daten	Datenermittlung und -Erfassung	Datenexploration durchführen Identifizieren von nützlichen Daten Sammeln von Daten	bei geeigneten und ungeeigneten Daten fällt es schwer solche auszuwählen, die ihre Annahme stützen (McNeill & Krajcik, 2007) Lernende haben oft nicht die Möglichkeit eigene Forschungsfragen zu formulieren; sie haben selten die Möglichkeit eigene Datenerhebungen zu planen und durchzuführen (Wolff et al., 2016)
	Evaluation und Sicherstellung der Qualität von Daten und ihrer Quellen	Kritische Bewertung von Datenquellen in Bezug auf ihre Glaubwürdigkeit Datensammlungen kritisch auf Fehler oder Probleme bewerten	Second Hand Data: unkritische Betrachtung der Daten, Daten werden nicht hinterfragt, gelten als "gegeben", es wird nicht über Messungenauigkeiten oder Fehlerquellen gesprochen (Hug & McNeill, 2008) Unter Anleitung sind Lernende in der Lage Gründe für Unterschiede in Messungen zu finden (Masnick & Klahr, 2003)
	Grundlegende Datenanalyse	Entwicklung eines Analyseplans Anwendung von Analysemethoden und -werkzeugen Durchführung explorativer Analysen Auswertung von Ergebnissen der Analyse Vergleich von Ergebnissen mit anderen Erkenntnissen	Schwierigkeiten repräsentative oder genaue Muster in Daten zu erkennen (Schauble, Glaser, Duschl, Schulz, & John, 1995) zu starre Vorgaben: Lernende haben oft nicht die Möglichkeit eigene Forschungsfragen zu formulieren; sie haben selten die Möglichkeit eigene Datenerhebungen zu planen und durchzuführen (Wolff et al., 2016) Informationsentnahme aus Diagrammen (Identifikation der Diagrammstruktur: Achsen, Skalen und Variablen erkennen); fehlende Lesestrategien (Probleme bei Erkennen von Trends, Vergleich von Werten, Lesen über die Daten hinaus) (Lachmayer et al., 2007) Kontext von Daten: Lernende brauchen eine didaktische Begleitung, um Zusammenhänge zwischen verschiedenen Daten zu erkennen (Kanari & Millar, 2004)
	Interpretation von Daten	Lesen und verstehen von Diagrammen, Tabellen und Grafiken Identifizierung der wichtigsten Erkenntnisse und Verknüpfung mit relevanten Informationen Unstimmigkeiten in den Daten finden	Informationsentnahme aus Diagrammen (Identifikation der Diagrammstruktur: Achsen, Skalen und Variablen erkennen); fehlende Lesestrategien (Probleme bei Erkennen von Trends, Vergleich von Werten, Lesen über die Daten hinaus) (Lachmayer et al., 2007) Problem bei Verknüpfung von Diagramm und Text (Lachmayer et al., 2007) Bei Schlussfolgerungen werden Messfehler oft nicht in Betracht gezogen – stattdessen basieren Annahmen auf Vorwissen (Masnick & Klahr, 2003; Gott & Duggan, 2007) Probleme zu verstehen, dass Daten Variationen zeigen & wann ist ein Beweis geeignet ist, um eine Frage wissenschaftlich zu beantworten (Germann & Aram, 1996) Schwierigkeiten Muster zu erkennen und eigene Schlüsse zu ziehen; Schlussfolgerungen waren häufiger fehlerhaft und nicht datengestützt (Hug & McNeill, 2008)
Datenauswertung	Identifizierung von Problemen mittels Daten	Verwendung von Daten, um Probleme in praktischen Situationen zu identifizieren (Effizienz am Arbeitsplatz) Verwendung von Daten, um Probleme auf höherer Ebene zu identifizieren (Politik, Umwelt, wissenschaftliche Experimente, Marketing, Wirtschaft,...)	First Hand Data: häufige Diskussion über Design, Messfehler und Ausführungsfehler; weniger inhaltliche Diskussion; keine Verbindung zu naturwissenschaftlichen Theorien oder Erklärungen Second Hand Data: unkritische Betrachtung der Daten, Daten werden nicht hinterfragt, gelten als "gegeben", es wird nicht über Messungenauigkeiten oder Fehlerquellen gesprochen; größere Schwierigkeiten Muster zu erkennen und eigene Schlüsse zu ziehen; Schlussfolgerungen waren häufiger fehlerhaft und nicht datengestützt (Hug & McNeill, 2008)
	Datenvisualisierung	Erstellung aussagekräftiger Grafiken zur Darstellung von Daten Beurteilung der Wirksamkeit von Grafiken Kritische Beurteilung der Genauigkeit und Eindeutigkeit von grafischen Auswertungen	Drei Probleme bei Verständnis von Diagrammen: Allgemeines Verständnis von Diagrammen, die Art des verwendeten Diagramms und inhaltliches Wissen über die Darstellung (McDermott, Rosenquist, & van Zee, 1987) Beschränkung auf das Erheben und einfache Auswerten von Daten („private claims“), kein kritisches hinterfragen oder Bezug zu größeren wissenschaftlichen Kontext („public claims“) (Gott & Duggan, 2007)
	Präsentieren von Daten	Bewertung, ob die grafische Darstellung geeignet für die Präsentation der Daten ist Bewertung, was die Ansprüche des Publikums sind und wie vertraut sie mit dem Thema sind Planung zur Art der Besprechung und der Präsentationsform Verwendung von aussagekräftigen Tabellen und Visualisierungen zur Vermittlung von Daten Präsentation von Argumenten und Ergebnissen in einer klaren und kohärenten Form	Fehlinterpretation von Diagrammen, Beispielsweise voreilige Trendannahme (Lachmayer et al., 2007) Persönliche Erfahrungen fließen in Erklärung von Phänomen mit ein (Warren, Ballenger, Ogonowski, Rosebery & Hudicourt-Barnes, 2001) Das ist abhängig vom individuellen Wissenstand (Gee, 1996; Moll, Amanti, Neff, & González, 1992) Schlussfolgerung aus eigenen Erfahrungen, Werten, Überzeugungen, statt Daten (Hogan & Maglienti, 2001) seltere Forderung in Schulen, dass Lernende ihre Datenauswertung systematisch und argumentativ verteidigen – die Daten „sprechen für sich“, statt kritisch geprüft zu werden (Gott & Duggan, 2007) fehlender Raum für offene Diskussion und kontroverse Auseinandersetzung mit Daten und deren Bedeutung (Newton et al., 1999)
	Datengestützte Entscheidungsfindung	Übersetzung gewonnener Daten/ Informationen in praktisch anwendbares Wissen Gewichtung von Informationen Abwägung der Vor- und Nachteile möglicher Lösungen Finden geeigneter Lösungen / Entscheidungen und deren Umsetzung	Bei geeigneten und ungeeigneten Daten fällt es schwer solche auszuwählen, die ihre Behauptung stützen (McNeill & Krajcik, 2007) Schwierigkeiten repräsentative oder genaue Muster in Daten zu erkennen (Schauble, Glaser, Duschl, Schulz, & John, 1995) Second Hand Data: unkritische Betrachtung der Daten, Daten werden nicht hinterfragt, gelten als "gegeben", es wird nicht über Messungenauigkeiten oder Fehlerquellen gesprochen; größere Schwierigkeiten Muster zu erkennen und eigene Schlüsse zu ziehen; Schlussfolgerungen waren häufiger fehlerhaft und nicht datengestützt (Hug & McNeill, 2008) Datensätze in Schulen sind häufig wenig komplex und ungeeignet ein Verständnis der Auswertung großer Datenmengen zu entwickeln (Wolff et al., 2016)

Abbildung 1: Auswahl von Schlüsselqualifikationen, die für das DataLab relevant sind, aus der Data Literacy Kompetenz Matrix nach Ridsdale et al. (2015), den darin enthaltenen Kompetenzen, den daraus abgeleiteten Kenntnissen und Aufgaben, sowie den Problemen, die Lernende im Umgang damit haben.

Für die Schlüsselqualifikation *Sammeln von Daten* sind die Kompetenzen *Datenermittlung und -Erfassung* sowie die *Evaluation und Sicherstellung der Qualität von Daten sowie ihrer Quellen* wichtig. Krajcik et al. (2008) sehen bei den nötigen Kenntnissen über das *Identifizieren von nützlichen Daten* Probleme bei der Auswahl geeigneter Daten, die eine Annahme stützen können. Wolff et al. (2016) sehen bei der *Durchführung von Datenexplorationen* und dem *Sammeln von Daten* Probleme, dass Lernende häufig keine Möglichkeiten haben, eigene Datenerhebungen durchzuführen.

Bei der Kompetenz zur *Evaluation von Datenquellen* kommen Hug und McNeill (2008) zum Schluss, dass besonders im Fall bei der Verwendung von Second Hand Data die Daten nicht hinterfragt und als „gegeben“ angenommen werden. Masnick und Klahr (2003) argumentieren, dass Lernende unter didaktischer Anleitung jedoch durchaus in der Lage sind, Gründe für Messfehler benennen zu können.

Die Schlüsselqualifikation der *Datenauswertung* untergliedert sich in die Kompetenzen *Grundlegende Datenanalyse, Interpretation von Daten, Identifizierung von Problemen mittels Daten, Datenvisualisierung, Präsentieren von Daten und datengestützte Entscheidungsfindung*. In der Kompetenz zur *Grundlegenden Datenanalyse* sehen Schauble et al. (1995) Schwierigkeiten, repräsentative oder genaue Muster in Daten zu erkennen, was wiederum nötig wäre, um Analysen auszuwerten und die Ergebnisse davon einordnen zu können. Wolff et al. (2016) berichten von häufig zu starren Vorgaben bei der Bearbeitung von Aufgaben, was verhindert, dass Lernende selbst eigene Forschungsfragen formulieren um damit eigene Datenerhebungen planen und durchführen können. Das steht der Entwicklung eines eigenen Analyseplans und der Durchführung explorativer Analysen entgegen. Der Vergleich eigener Erkenntnisse mit den Ergebnissen anderer Untersuchungen scheitert laut Lachmayer et al. (2007) an fehlenden Lesestrategien von Diagrammen (Erkennen von Trends, Vergleich von Werten und „Lesen über die Daten hinaus“).

Bei der Kompetenz der *Interpretation von Daten* stehen Fertigkeiten im Fokus, die ermöglichen, dass Diagramme, Tabellen und Grafiken gelesen werden können. Dabei müssen wichtige Erkenntnisse in den Darstellung identifiziert und Unstimmigkeiten in den Daten ausfindig gemacht werden können. Die Lernenden haben laut Lachmayer et al. (2007) häufig Probleme bei der Informationsentnahme aus Diagrammen (Identifikation der Diagrammstruktur: Achsen, Skalen und Variablen erkennen, Erkennen von Trends, Vergleich von Werten, Lesen über die Daten hinaus), sowie bei der Verknüpfung von Informationen aus Abbildung und Textform. Messfehler werden häufig nicht in Betracht gezogen oder thematisiert, oder nur sehr oberflächlich behandelt (Gott & Duggan, 2007; Masnick & Klahr, 2003). Nach Germann und Aram (1996) und Hug und McNeill (2008) haben Lernende Probleme zu verstehen, dass Daten aufgrund von Messfehlern und Messungenauigkeiten variieren können. Zudem zeigen Lernende größere Schwierigkeiten, wenn es darum geht Muster in Daten zu erkennen und eigene Schlussfolgerungen zu erstellen. Schlussfolgerun-

gen können überdies häufig fehlerhaft und nicht Daten gestützt erfolgen.

Bei der Kompetenz *Identifizierung von Problemen mittels Daten* sollen Lernende in der Lage sein, Probleme auf der Grundlage von vorliegenden Daten zu identifizieren. Hierbei kann es sich sowohl um Probleme in praktischen Situationen handeln (z.B. Effizienz am Arbeitsplatz) oder um Probleme auf einer übergeordneten Ebene (wie z.B. Umwelt, Politik, wissenschaftlichen Experimenten, ...). Hier kommt es nach den Erkenntnissen von Hug und McNeill (2008) zu Problemen aus der Verwendung von First Hand Data (selbst erhobenen Daten) und Second Hand Data (bereitgestellte Daten). Lernende stellen bei First Hand Data seltener eine Verbindung zu naturwissenschaftlichen Theorien oder Erklärungen her. Bei Second Hand Data waren die Schlussfolgerungen häufiger fehlerhaft und nicht Daten gestützt. Beides verhindert, dass Daten dazu beitragen, Probleme in den jeweiligen Situationen identifizieren zu können.

Die Kompetenz der *Daten Visualisierung* enthält, dass die Lernenden aussagekräftige Grafiken zur Darstellung von Daten erstellen können. Sie sollten beurteilen können, wie effektiv die Grafiken die Informationen vermitteln können und sie sollten in der Lage sein die eigenen Grafiken kritisch auf ihre Genauigkeit und ihre Eindeutigkeit hin zu bewerten. Wie McDermott et al. (1987) und Gott und Duggan (2007) zeigen, haben Lernende Probleme bei dem grundsätzlichen Verständnis von Diagrammen und der Auswahl einer geeigneten Diagrammart. Sie verfügen über wenig inhaltliches Wissen über die vorhandenen Darstellungsmöglichkeiten. Wenn sie Diagramme erstellen, werden die Daten in einfacher Form ausgewertet, werden wenig kritisch hinterfragt oder nicht in Bezug zu einem größeren wissenschaftlichen Kontext gesetzt (Gott & Duggan, 2007).

Bei der *Präsentation von Daten* sollten Lernende in der Lage sein, eine Bewertung vorzunehmen, ob die gewählte Form der Darstellung geeignet und aussagekräftig genug ist und inwieweit das Publikum bereits mit der Thematik vertraut ist. Es sollten aussagekräftige Tabellen und Grafiken gewählt werden und die Präsentation sollte in einer klaren und kohärenten Form erfolgen. Lernende haben Schwierigkeiten mit der Interpretation von Diagrammen, wobei es zu voreiligen Trendannahmen kommt, persönliche Erfahrungen, der individuelle Wissensstand, sowie eigene Werte und Überzeugungen mit in die Erklärung von Phänomenen einfließen (Gee, 1996; Hogan & Maglienti, 2001; Lachmayer et al., 2007; Warren et al., 2001). Gott und Duggan (2007) und Newton et al. (1999) sehen, dass es an Schulen nur selten dazu kommt, dass die Datenauswertung systematisch und argumentativ verteidigt werden muss (statt dessen verfolgen die Lernenden den Ansatz, dass Daten „für sich sprechen“) und, dass es wenig Raum für offene Diskussionen und eine kontroverse Auseinandersetzung mit Daten und deren Bedeutung gibt.

Für die von Ridsdale et al. (2015) erwähnte Kompetenz der *datengestützten Entscheidungsfindung*, sollten Lernende in der Lage sein die gewonnenen Daten bzw. Informationen in praktisch anwendbares Wissen zu übersetzen, indem sie ihr Urteilsvermögen nutzen, um

Informationen zu gewichten und die Vor- und Nachteile möglicher Lösungen gegeneinander abzuwägen und schließlich geeignete Lösungen bzw. Entscheidungen zu treffen und umzusetzen. Wolff et al. (2016) sehen ein Problem darin, dass Daten an Schulen häufig wenig komplex sind und folglich ungeeignet dafür sind, ein Verständnis für die Auswertung großer Datenmengen zu entwickeln. McNeill und Krajcik (2007), Schable et al. (1995) und Hug und McNeill (2008) sehen viele Probleme bei der Auswertung von Daten: Lernenden fällt es schwer aus den Daten solche auszuwählen, die ihre Annahmen stützen. Sie haben Schwierigkeiten repräsentative oder genaue Muster in den Daten zu erkennen. Zudem werden Daten teils nicht auf mögliche Messfehler hin hinterfragt. Schlussfolgerungen sind demnach häufig fehlerhaft und nicht datengestützt. All diese Aspekte erschweren es, aus den Daten konkrete Entscheidungen abzuleiten und geeignete Lösungen zu entwickeln.

2.2 Didaktische Methoden zur Förderung von Data Literacy und Kriterien zur Visualisierung von Umweltdaten

Um den dargelegten Problemen im Zusammenhang mit der Entwicklung von Kompetenzen im Bereich Data Literacy zu begegnen werden unterschiedliche didaktische Methoden vorgeschlagen, die in Tabelle 2 zur Übersicht dargestellt werden. Diese didaktischen Methoden können bei der Entwicklung der Lerneinheiten im *Humboldt Explorers* Projekt zum Einsatz kommen. Eine ausführliche Darstellung und Diskussion der didaktischen Methoden ist nicht Gegenstand der vorliegenden Arbeit, da der Schwerpunkt in der Entwicklung einer integrierten webbasierten Visualisierung von Umweltdaten im Kontext von Data Literacy besteht. In diesem Zusammenhang sollen insbesondere die Kriterien zur Visualisierung von Umweltdaten im Bildungskontext berücksichtigt werden, die aus den in Tabelle 3 aufgeführten Studien abgeleitet wurden.

Tabelle 2: Didaktische Methoden zur Förderung von Data Literacy, abgeleitet aus ausgewählten Studien.

Didaktische Methode	Beschreibung	Quelle(n)
Vorwissen einbeziehen, um Fehlinterpretation zu vermeiden	Vorhandene Alltagsvorstellungen nutzen, um Missverständnisse zu reduzieren und an bestehendes Wissen anzuknüpfen.	Gott und Duggan (2007); Wolff et al. (2016)
Verknüpfung von Alltagserfahrungen mit wissenschaftlichem Denken	Nutzung von Erfahrungswissen der Lernenden als Ressource für das Verständnis von Daten und wissenschaftlichen Konzepten.	Warren et al. (2001)
Interdisziplinäre Lerneinheiten	Verknüpfung unterschiedlicher Fachperspektiven, um Data Literacy über Fächergrenzen hinweg zu fördern.	Swan et al. (2009)
Kombination von First & Second-Hand Data	Nutzung sowohl selbst erhobener als auch bestehender Datenquellen zur Förderung kritischer Reflexion und Validierung.	Hug und McNeill (2008)
Iterative Anwendung des fragengesteuerten Lernens (PPDAC-Modell)	Strukturierte Analyse durch wiederholte Anwendung des Plan-Problem-Data-Analysis-Conclusion Modells.	Wolff et al. (2016)
Diagramme lesen, konstruieren und integrieren	Förderung von Visual Literacy durch aktives Arbeiten mit unterschiedlichen Diagrammtypen und Datenvisualisierungen.	Lachmayer et al. (2007)
Hands-On (interaktiv)	Explorative Auseinandersetzung mit Daten durch direkte Experimente und sensorbasiertes Arbeiten.	Wolff et al. (2016)
Data Storytelling	Aufbereitung von Daten in narrativer Form, um das Verstehen und Kommunizieren von Ergebnissen zu fördern.	Wolff et al. (2016)
Daten präsentieren	Förderung der Kommunikationskompetenz durch Präsentation eigener Ergebnisse.	Ridsdale et al. (2015)
Einsatz von Technologie	Verwendung digitaler Tools zur Erfassung, Verarbeitung und Darstellung von Daten.	Wolff et al. (2016); Ridsdale et al. (2015)
Kritische Auseinandersetzung mit Datenquellen	Reflexion über Qualität und Herkunft von Daten.	Wolff et al. (2016); Gott und Duggan (2007)
Stufenweise Einbettung von realitätsnahen Fallbeispielen	Aufbau komplexer Datenkompetenz durch sukzessive Einführung realistischer Problemstellungen.	Hug und McNeill (2008); Wolff et al. (2016)
Über Messfehler, Grenzen und Unsicherheiten nachdenken und diskutieren	Förderung kritischen Denkens über Messgenauigkeit und Unsicherheiten im Umgang mit realen Daten.	Masnick und Klahr (2003); Gott und Duggan (2007)
Arbeit mit realen und komplexen Datensätzen	Nutzung echter, komplexer Datensätze zur realitätsnahen Schulung von Analyse- und Bewertungskompetenzen.	Wolff et al. (2016)
Verwendung von Live-Daten	Erhöhung der Relevanz und Aktualität der Lernprozesse durch die Arbeit mit aktuellen Datenquellen.	Wolff et al. (2016)
Feldforschung, Diagnoseaufgaben, explorative und mehrdimensionale Datensituationen	Förderung von Inquiry-Kompetenzen durch komplexe, offene und mehrperspektivische Datenerhebungen.	Wolff et al. (2016); Hug und McNeill (2008)

2.3 Zielsetzung & Forschungsfrage

Ziel der vorliegenden Arbeit ist es, Module zur Visualisierung von Umweltdaten für das DataLab zu konzipieren, die gezielt zur Förderung von Data Literacy beitragen. Dabei sollen die bekannten Probleme (siehe Abschnitt 2.1) im Umgang mit Daten berücksichtigt und die identifizierten Kriterien (siehe Tabelle 3) für die Gestaltung von Umweltdatavisualisierungen systematisch in die Entwicklung der Module integriert werden.

Wie lässt sich eine webbasierte Plattform zur interaktiven Darstellung von Umweltdaten gestalten, um ausgewählte Komponenten der Data Literacy zu fördern?

3 Material und Methoden

3.1 Konzeption von Modulen zur Visualisierung von Umweltdaten für das DataLab

3.1.1 Konzeption: Messstellen-Karte

Ziel: Das Modul soll die die Kriterien zur Visualisierung von Umweltdaten *Repräsentationsvielfalt, Interpretierbarkeit, Kontextualisierung, Sichtbarmachung der Datenqualität, mehrdimensionale Darstellung, sowie Ästhetik und Benutzerfreundlichkeit* umsetzen.

Beschreibung: Auf einer interaktiven Landkarte sollen alle Messstellen verzeichnet werden, an denen Umweltdaten gesammelt werden. Die Karte soll verschiebbar und vergrößerbar gestaltet werden. Wird eine Messstation ausgewählt, sollen alle verfügbaren Messdaten eines Standorts im Verlauf der letzten Tage angezeigt werden. Zusätzlich sollen Satellitenbilder und 3D-Modelle von Gebäuden neben der Straßendarstellung eingeblendet werden können. Es soll eine Funktion eingebaut werden, die den eigenen Standort bestimmt und automatisch auf ihn hin vergrößert. Die Darstellung der Diagramme soll sich an die durch Konvention festgelegten Vorgaben anlehnen, wie sie von Lachmayer et al. (2007) beschrieben werden. Dabei sollen die Messdichte, fehlende Werte sowie – sofern verfügbar – gesetzlich festgelegte Grenzwerte in der Darstellung berücksichtigt werden. Die Daten sollen in 5-Minuten-Intervallen geglättet werden (die einzelnen Werte werden auf volle 5 Minuten gerundet und anschließend wird der Median für alle Werte an diesen Zeitpunkten berechnet), alle X-Achsen sollen einheitlich auf dieselben Zeitintervalle abgestimmt werden. Jedes Diagramm soll eine eigene Farbe erhalten, um das Ablesen zu erleichtern. Die Diagramme sollen interaktiv nutzbar sein: Bereiche sollen auswählbar und vergrößerbar sein, einzelne Messwerte sollen bei Auswahl direkt neben dem Graphen angezeigt werden. Sind die Messwerte unzusammenhängend, soll die Darstellung als Streudiagramm erfolgen, andernfalls soll ein Liniendiagramm verwendet werden.

Theoretischer Bezug zu Data Literacy: Durch die Darstellung der Sensorstandorte auf

Tabelle 3: Kriterien zur Visualisierung von Umweltdaten im Kontext Data Literacy, abgeleitet aus ausgewählten Studien.

Kriterium	Beschreibung	Quelle(n)
Repräsentationsvielfalt	Lernende sollen zwischen verschiedenen Darstellungen wechseln können, um Daten kontextgerecht zu interpretieren.	Ridsdale et al. (2015); Lachmayer et al. (2007)
Interpretierbarkeit	Visualisierungen müssen verständlich und zielgruppengerecht sein, ohne zu starke Vereinfachung.	Wolff et al. (2016); Lachmayer et al. (2007)
Kontextualisierung	Daten sollen räumlich (etwa per Karte) oder zeitlich (etwa per Zeitachse) eingeordnet werden.	Hug und McNeill (2008); Gott und Duggan (2007)
Handlungsrelevanz	Visualisierungen sollen reale Entscheidungsprozesse unterstützen oder simulieren.	Wolff et al. (2016); Ridsdale et al. (2015)
Interaktivität / Exploration	Visualisierungen sollen zur aktiven Erkundung anregen, etwa durch Filter, Layer oder Zoom.	Wolff et al. (2016); Ridsdale et al. (2015)
Bezug zu eigenen Daten	Visualisierungen sollen auch eigene Erhebungen oder Messwerte der Lernenden abbilden können.	Wolff et al. (2016); Hug und McNeill (2008)
Vergleich und Mustererkennung	Visualisierungen sollen den Vergleich von Standorten / Zeitpunkten und das Erkennen von Trends ermöglichen.	Lachmayer et al. (2007); Hug und McNeill (2008)
Datenqualität sichtbar machen	Unsicherheiten, Messfehler oder Lücken sollen in der Visualisierung explizit kenntlich gemacht werden (z.B. spezielle Symbole oder Farbgestaltung).	Gott und Duggan (2007); Wolff et al. (2016)
Mehrdimensionale Darstellung	Wenn sinnvoll, sollen mehrere Variablen kombiniert dargestellt werden können (z.B. bivariate Karten, kombinierte Diagramme).	Hug und McNeill (2008); Wolff et al. (2016)
Ästhetik und Benutzerfreundlichkeit	Die grafische Gestaltung soll ansprechend, übersichtlich und intuitiv verständlich sein (z.B. Farbschemata, klare Achsenbeschriftungen).	Ridsdale et al. (2015)
Adaptierbarkeit	Visualisierungen sollen flexibel anpassbar sein, etwa an verschiedene Fragestellungen oder Zielgruppen.	Wolff et al. (2016)

einer Karte wird Lernenden ein direkter Bezug zur räumlichen Herkunft der Daten ermöglicht, wodurch die Kompetenz *Verständnis der Datenherkunft und Kontextualisierung* adressiert wird. Auf diese Weise wird nicht nur das Verständnis darüber gefördert, wo Daten erhoben wurden, sondern es werden auch Rückschlüsse auf die jeweiligen Umgebungsbedingungen ermöglicht, unter denen die Messungen durchgeführt wurden (Hug & McNeill, 2008; Wolff et al., 2016). Mit dem Modul soll zudem die Kompetenz *Interpretation von Zeitreihen und Diagrammen* gestärkt werden, indem die Fähigkeiten zum Ablesen von Diagrammen, zum Vergleichen von Messwerten und zum Erkennen von Mustern in Daten gefördert werden (Lachmayer et al., 2007; Ridsdale et al., 2015). Die Kompetenz *Umgang mit Messunsicherheiten und Messlücken* wird durch die Darstellung fehlender Werte sowie durch die Kombination von Linien- und Streudiagrammen unterstützt, wodurch das Verständnis von Datenqualität und Messfehlern verbessert wird (Masnick & Klahr, 2003). Durch die visuelle Hervorhebung von Schwellwertüberschreitungen gesetzlicher Grenzwerte wird die gesellschaftliche Relevanz der Daten verdeutlicht und eine ethische Bewertung der Messreihen ermöglicht, was zur Förderung der *ethischen Kompetenz* beiträgt (Wolff et al., 2016).

Förderung von Teilkompetenzen aus dem Bereich Data Literacy nach Ridsdale et al. (2015): Zur Förderung der Kompetenz *Interpretation von Daten* wird vorgesehen, Daten auch bei unregelmäßigen Messreihen darzustellen. Bei fehlenden Werten soll die Darstellung automatisch von Linien- auf Streudiagramme wechseln. Dadurch wird eine Visualisierung der Werte ermöglicht, auch wenn einzelne Messungen fehlen, was den Umgang mit unvollständigen Datensätzen und Unsicherheiten unterstützt sowie für Datenlücken sensibilisiert. Durch die Anzeige der Standorte auf einer Karte wird die Herkunft der Daten sowie der Kontext, in dem sie erhoben wurden, sichtbar gemacht, was die Kompetenz *Evaluation und Sicherstellung der Qualität von Daten und ihren Quellen* stärkt.

Allgemeine Lernziele: Lernende werden in die Lage versetzt, den geografischen Standort mit den gemessenen Umweltparametern in Verbindung zu setzen. Dadurch entwickeln sie ein Verständnis dafür, wie unterschiedliche Umgebungsbedingungen – wie Bebauung, Vegetation oder städtische Infrastruktur – die erhobenen Messwerte beeinflussen können. Zusätzlich wird die Fähigkeit gefördert, räumliche Zusammenhänge zwischen Messdaten und deren Entstehungskontext kritisch zu reflektieren.

Mögliche User Story: Lernende erhalten die Möglichkeit, eigenständig zu erforschen, welche Umweltdaten an den jeweiligen Standorten erhoben werden. Dabei kann entweder ein eigener Forschungsauftrag entwickelt oder ein vorgegebener Arbeitsauftrag durch die Lehrkraft bearbeitet werden. Beispielsweise können die an verschiedenen Standorten gemessenen Temperaturen untersucht und miteinander verglichen werden. Die gewonnenen Erkenntnisse können anschließend mit den jeweiligen Standortbedingungen in Beziehung gesetzt und hinsichtlich möglicher Einflussfaktoren reflektiert werden.

3.1.2 Konzeption: Hexagonale Dichtekarte

Ziel: Hier sollen die Kriterien zur Visualisierung von Umweltdaten *Kontextualisierung* und *Datenqualität sichtbar machen* umgesetzt werden.

Beschreibung: Auf einer Karte sollen Umweltdaten durch sogenanntes Binning, also das Zusammenfassen von Messreihen in sechseckigen Flächenbereichen, visualisiert werden. Diese Darstellungsform soll eine übermäßige Über- oder Unterrepräsentation vermeiden, wie sie bei herkömmlichen Flächenkartogrammen häufig auftritt. Vorgesehen ist eine moderne und abstrahierte Visualisierung, die eine repräsentative Darstellung der Daten für bestimmte Gebiete ermöglicht. Durch die gewählte Darstellungsweise soll eine konsistente Kartenansicht geschaffen, die Lesbarkeit optimiert und eine explorative Datenanalyse über Raum und Zeit hinweg erleichtert werden. Zur Umsetzung dieses Ziels sollen die Datenreihen über einen Zeitraum der letzten 48 Stunden gesammelt und in animierter Form dargestellt werden.

Theoretischer Bezug zu Data Literacy: Die hexagonale Dichtekarte stellt eine Form komplexer Visualisierung dar, wie sie von Wolff et al. (2016) zur Förderung von Data Literacy empfohlen wird. Durch diese Darstellungsform wird es ermöglicht, nahezu beliebige Datensätze in ihrem zeitlichen Verlauf auf einer Karte abzubilden. Auf diese Weise kann ein tiefergehendes Verständnis für räumlich-zeitliche Zusammenhänge entwickelt werden, das mit herkömmlichen Darstellungen, wie Balken-, Streu- oder Liniendiagrammen, nicht in gleicher Weise erreicht werden könnte.

Förderung von Teilkompetenzen aus dem Bereich Data Literacy nach Ridsdale et al. (2015): Nach der Kompetenzmatrix von Ridsdale et al. (2015) werden durch diese Visualisierungsform mehrere Teilkompetenzen gefördert. Durch die Verortung und Sammlung von Sensorwerten auf einer realen Karte sowie die Kontextualisierung der Daten über Raum und Zeit wird insbesondere die Kompetenz der *Datensammlung* unterstützt, da ein wesentliches Element dieser Kompetenz das Verständnis des Entstehungskontextes von Daten darstellt. Zudem soll die Kompetenz der *Auswertung von Daten* gefördert werden, da die Einschätzung der Datenqualität eine zentrale Rolle im Analyseprozess einnimmt. Schließlich trägt die explorative Arbeit mit der Karte, die Nutzung von Farbskalen sowie die räumlich-zeitliche Darstellung zur Förderung der Kompetenz *Interpretation von Daten* bei.

Allgemeine Lernziele: Durch diese Darstellungsform sollen Lernende befähigt werden, ein Verständnis für die räumliche Verteilung von Daten zu entwickeln. Auf dieser Grundlage sollen sie in die Lage versetzt werden, Zusammenhänge zwischen Messwerten und geographischen Gegebenheiten kritisch zu analysieren sowie eine fundierte Bewertung der Datenqualität vorzunehmen.

Mögliche User Story: Lernende erhalten die Möglichkeit, durch die Analyse der Karte

eigenständig zu erkunden, wie sich Umweltmesswerte räumlich verteilen. Sie können Regionen mit auffälligen Mustern identifizieren, Hypothesen über die Ursachen dieser Verteilungen entwickeln und die Qualität sowie Aussagekraft der zugrunde liegenden Daten kritisch hinterfragen. Dabei werden sie dazu angeregt, räumliche Zusammenhänge zwischen Umweltfaktoren und gemessenen Parametern eigenständig zu erkennen und zu bewerten.

3.1.3 Konzeption: Berliner Wärmeinseleffekt

Ziel: Das Modul soll die Kriterien zur Visualisierung von Umweltdaten *Kontextualisierung* und die *mehrdimensionale Darstellung* umsetzen.

Beschreibung: Zur gemeinsamen Darstellung der Werte für Lufttemperatur und Wind soll eine Karte in einer möglichst intuitiv verständlichen Form bereitgestellt werden. Hierfür ist die Verwendung eines bivariaten Flächendiagramms vorgesehen, bei dem die Flächen in Abhängigkeit von der Ausprägung beider Werte eingefärbt werden. Treten beide Werte gleichzeitig auf, sollen die Farben für Temperatur und Wind miteinander kombiniert werden. Darüber hinaus sollen die Werte für verschiedene Zeitpunkte dargestellt werden, wobei entweder eine statische Anzeige für ausgewählte Uhrzeiten oder eine Animation über den Tagesverlauf gewählt werden kann.

Theoretischer Bezug zu Data Literacy: In der Arbeit von Wolff et al. (2016) wird betont, dass komplexe Visualisierungen und Datensätze gezielt zur Förderung von Data Literacy eingesetzt werden sollten. Es wird hervorgehoben, dass traditionelle Darstellungen, wie einfache Linien- oder Streudiagramme, nicht ausreichen, um Lernende auf moderne Formen der Datenvisualisierung vorzubereiten, die zunehmend ästhetische, interaktive oder künstlerische Elemente integrieren. Daher wird der Einsatz anspruchsvollerer Darstellungen im Unterricht empfohlen, um die Analysek Kompetenz sowie das kritische Denken der Lernenden zu stärken. Die Arbeit mit Second-Hand-Daten wird zudem als förderlich für die kritische Betrachtung von Datenquellen, Messgenauigkeiten und potenziellen Fehlerquellen beschrieben. Hierdurch werden die Fähigkeiten unterstützt, Muster in Daten zu erkennen und eigenständig Schlussfolgerungen zu ziehen (Hug & McNeill, 2008).

Förderung von Teilkompetenzen aus dem Bereich Data Literacy nach Ridsdale et al. (2015): Durch diesen Ansatz werden nach Ridsdale et al. (2015) mehrere Teilkompetenzen im Bereich Data Literacy gefördert. Dazu zählen die Kompetenz *Interpretation von Daten*, die die Verifizierung zentraler Erkenntnisse sowie die Verknüpfung mit anderen relevanten Informationen und das Erkennen von Unstimmigkeiten in Datensätzen umfasst, die Kompetenz *Identifizierung von Problemen mittels Daten*, bei der Daten genutzt werden, um Probleme auf höherer Ebene zu erkennen, sowie die Kompetenz *datengestützte Entscheidungsfindung*, die die Abwägung möglicher Implikationen unterschiedlicher Handlungsoptionen einschließt.

Allgemeine Lernziele: Mit Hilfe dieser Karte sollen Lernende in die Lage versetzt wer-

den, ein grundlegendes Verständnis für den Berliner Wärmeinseleffekt zu entwickeln. Dabei sollen sie befähigt werden, eigene Hypothesen über die Entstehung und Verteilung von Wärmeinseln zu formulieren sowie Theorien über deren Ursachen zu entwickeln und kritisch zu hinterfragen. Durch die Arbeit mit den visualisierten Daten wird zudem das analytische Denken im Umgang mit raumbezogenen Umweltphänomenen gefördert.

Mögliche User Story: Lernende erhalten die Möglichkeit, anhand der Karte eigenständig die Verteilung von Temperatur- und Winddaten in Berlin zu erkunden. Sie können eigene Hypothesen zur Entstehung von Wärmeinseln entwickeln, diese anhand der visualisierten Daten überprüfen und Zusammenhänge zwischen Bebauungsdichte, Vegetation und Temperaturverläufen erkennen. Dabei werden sie dazu angeregt, Ursachen für beobachtete Muster kritisch zu hinterfragen und eigene Erklärungsansätze auf Grundlage der erhobenen Umweltdaten zu formulieren.

3.1.4 Konzeption: Live vor Ort - Sensordaten am Standort

Ziel: Dieses Modul soll folgende Kriterien zur Visualisierung von Umweltdaten umsetzen: *Repräsentationsvielfalt, Kontextualisierung, Handlungsrelevanz, Bezug zu eigenen Daten, Datenqualität sichtbar machen, mehrdimensionale Darstellung, sowie Ästhetik und Benutzerfreundlichkeit.*

Beschreibung: Mit diesem Modul soll es ermöglicht werden, aktuelle Messwerte aller SenseBoxen eines Standorts, gruppiert über den GroupTag *Humboldt Explorers*, in Form von Linien- und Streudiagrammen zu sammeln und darzustellen. Auf diese Weise soll eine Live-Visualisierung standortbezogener Umweltdaten innerhalb einer interaktiven Unterrichtseinheit realisiert werden. Über eine zusätzlich eingeblendete Standortkarte soll eine Kontextualisierung der Messdaten mit der jeweiligen Umgebung ermöglicht werden. Die Darstellung der Diagramme soll wahlweise gesammelt für alle SenseBoxen aller Gruppen oder aufgeschlüsselt nach einzelnen Gruppen erfolgen. Trotz der komplexen Visualisierungsform wird eine einfache und intuitiv bedienbare Nutzeroberfläche angestrebt.

Theoretischer Bezug zu Data Literacy: In ihrer Arbeit betonen Wolff et al. (2016) die Bedeutung von zeitlich aktuellen Abfragen (Echtzeitdaten), um das Verständnis für die Dynamik von Daten sowie für die Aktualität und Frequenz von Datenströmen zu fördern. Hug und McNeill (2008) schlagen zudem die kombinierte Nutzung von First- und Second-Hand-Daten vor, um ein vertieftes Verständnis für Datenerhebung und -nutzung zu ermöglichen. Durch dieses Modul wird es Lernenden ermöglicht, selbstständig und in großem Umfang Daten zu erheben und diese automatisch zu visualisieren. Hierdurch werden laut Hug und McNeill (2008) Kompetenzen wie das Verständnis von Messdesigns, das Erkennen von Messfehlern und Ausführungsfehlern sowie die kritische Reflexion über deren Ursachen gefördert. Zudem wird den Lernenden die Möglichkeit eröffnet, eigene Forschungsfragen zu entwickeln und eigenständige Datenerhebungen zu planen und durchzuführen

(Wolff et al., 2016).

Förderung von Teilkompetenzen aus dem Bereich Data Literacy nach Ridsdale et al. (2015): Nach Ridsdale et al. (2015) werden durch dieses Modul mehrere Teilkompetenzen im Bereich Data Literacy gefördert. Im Bereich der Schlüsselqualifikation *Sammeln von Daten* werden insbesondere die Kompetenzen *Datenermittlung und -erfassung* sowie *Evaluation und Sicherstellung der Qualität von Daten und ihren Quellen* adressiert. Lernende werden in die Lage versetzt, eigenständig Daten zu explorieren, zu sammeln und deren Quellen kritisch im Hinblick auf Glaubwürdigkeit und potenzielle Fehler zu bewerten. Darüber hinaus wird die Kompetenz *Interpretation von Daten* gefördert, indem Lernende befähigt werden, zentrale Erkenntnisse aus den erhobenen Daten zu identifizieren, diese mit weiteren relevanten Informationen zu verknüpfen und Unstimmigkeiten im Datensatz zu erkennen. Zudem trägt die Möglichkeit der gleichzeitigen Darstellung aller Messwerte unterschiedlicher Gruppen zur Förderung der Kompetenz *Präsentation von Daten* bei, da Lernende ihre Ergebnisse angemessen aufbereiten und kommunizieren können.

Allgemeine Lernziele: Lernende sollen befähigt werden, komplexe ökologische Zusammenhänge sowie Wechselwirkungen zwischen verschiedenen Umweltfaktoren an einem konkreten Standort in Berlin zu beschreiben und zu erläutern. Dabei sollen sie in die Lage versetzt werden, Wechselbeziehungen kritisch zu analysieren, Auswirkungen einzelner Faktoren auf das Gesamtsystem zu erkennen und daraus fundierte Erklärungsansätze für beobachtete Phänomene abzuleiten.

Mögliche User Story: Lernende erhalten die Möglichkeit, durch die Analyse von Live-Umweltdaten komplexe ökologische Zusammenhänge an einem konkreten Berliner Standort zu erforschen. Sie können eigenständig Wechselwirkungen zwischen verschiedenen Umweltfaktoren erkennen, Hypothesen über deren Ursache-Wirkungs-Beziehungen entwickeln und eigene Erklärungsansätze für beobachtete Phänomene formulieren. Dabei werden sie angeregt, sowohl einzelne Messwerte als auch deren räumlichen und zeitlichen Kontext kritisch zu betrachten und in ihre Analyse einzubeziehen.

3.1.5 Konzeption: Live im Vergleich - Sensordaten über Standorte hinweg

Ziel: Hier sollen die Kriterien zur Visualisierung von Umweltdaten *Repräsentationsvielfalt, Kontextualisierung, Handlungsrelevanz, Interaktivität/Exploration, Bezug zu eigenen Daten, Vergleich und Mustererkennung, Datenqualität sichtbar machen, mehrdimensionale Darstellung, sowie Ästhetik und Benutzerfreundlichkeit* umgesetzt werden.

Beschreibung: In Anlehnung an das Modul *Live vor Ort - Sensordaten am Standort* sollen auch in diesem Modul Live-Messdaten in Diagrammen dargestellt werden. Im Unterschied zur bisherigen Darstellung wird hier ermöglicht, beliebige Standorte und damit beliebige SenseBoxen miteinander zu vergleichen. Ziel ist es, Lernenden die Möglichkeit zu eröffnen, unterschiedliche Standorte parallel anzuzeigen, um Zusammenhänge zwischen Um-

weltbedingungen – etwa deren Einfluss auf Temperaturwerte – eigenständig zu erkennen und zu analysieren.

Theoretischer Bezug zu Data Literacy: Hug und McNeill (2008) betonen in ihrer Arbeit die Bedeutung des wechselseitigen Einsatzes von First- und Second-Hand-Daten, um über First-Hand-Daten Diskussionen zu Versuchsdesign, Messfehlern und Durchführung anzuregen sowie die kritische Auseinandersetzung mit der Herkunft von Daten zu fördern. Durch die Einbindung von Second-Hand-Daten sollen darüber hinaus vertiefte inhaltliche Diskussionen ermöglicht werden. Auch in diesem Modul werden Live-Daten abgefragt und in Diagrammen dargestellt. Auf diese Weise wird die Kontextualisierung von Daten aus dem direkten Lebensumfeld unterstützt, wodurch eine unmittelbare Greifbarkeit der Daten erreicht wird. Die Möglichkeit, Veränderungen direkt zu beobachten, fördert die Mustererkennung sowie das Verständnis für dynamische Prozesse. Die Nutzung von Echtzeitdaten erhöht zudem die Authentizität der Lernsituation, indem sie den Eindruck von „echter Wissenschaft“ oder „realer Welt“ vermittelt und somit Lernmotivation und Relevanz steigert. Darüber hinaus werden Lernende durch den Umgang mit Live-Daten dazu angeregt, eigene Fragestellungen zu entwickeln und diese unmittelbar zu überprüfen. Die Arbeit mit Live-Daten unterstützt somit nicht nur die Förderung eigenständiger Explorativen, sondern trägt auch zur Entwicklung von Kompetenzen bei, die für eine aktive digitale Teilhabe in einer zunehmend datenbasierten Gesellschaft erforderlich sind (Wolff et al., 2016).

Förderung von Teilkompetenzen aus dem Bereich Data Literacy nach Ridsdale et al. (2015): Durch dieses Modul werden nach der Kompetenzmatrix von Ridsdale et al. (2015) mehrere Teilkompetenzen im Bereich Data Literacy gefördert. Im Bereich der *Datenermittlung und -erfassung* wird Lernenden ermöglicht, im Rahmen explorativer Analysen die Daten einer zur Verfügung gestellten SenseBox direkt anzeigen zu lassen. Darüber hinaus werden die Kompetenzen *Evaluation und Sicherstellung der Qualität von Daten und ihren Quellen* sowie *Grundlegende Datenanalyse* adressiert, da beliebig viele externe Datenquellen einbezogen und direkte Vergleiche zwischen unterschiedlichen Messreihen und Standorten ermöglicht werden. Durch die aktive Nutzung und Betrachtung der Diagramme wird zudem die Kompetenz *Interpretation von Daten* gefördert, indem Lernende angeregt werden, Diagramme auszuwerten und Unstimmigkeiten in den Datensätzen kritisch zu reflektieren.

Allgemeine Lernziele: Lernende sollen befähigt werden, im Rahmen einer selbst definierten oder vorgegebenen Forschungsfrage Live-Datensätze unterschiedlicher Standorte abzurufen, darzustellen und gezielt miteinander zu vergleichen. Dabei sollen sie in die Lage versetzt werden, eigenständig Muster, Zusammenhänge und Unterschiede in den Messreihen zu identifizieren und diese in Bezug auf ihre Forschungsfrage kritisch zu interpretieren.

Mögliche User Story: Lernende erhalten die Möglichkeit, eigene oder vorgegebene Forschungsfragen zu entwickeln und mithilfe von Live-Daten unterschiedlicher Standorte zu untersuchen. Sie wählen gezielt Standorte aus, rufen aktuelle Umweltdaten ab und vergleichen diese hinsichtlich relevanter Einflussgrößen. Dabei analysieren sie Muster, erkennen Unterschiede zwischen verschiedenen Standorten und ziehen eigenständig Schlussfolgerungen in Bezug auf ihre Forschungsfrage.

3.1.6 Konzeption: Interaktive Datenexploration

Ziel: Dieses Modul setzt die Kriterien zur Visualisierung von Umweltdaten *Interpretierbarkeit, Kontextualisierung, Interaktivität/Exploration*, sowie *Adaptierbarkeit* um.

Beschreibung: In diesem Modul soll Lernenden die Möglichkeit gegeben werden, eigene Visualisierungen auf Basis aktueller Sensordaten zu erstellen. Über eine interaktive Benutzeroberfläche können Parameter wie die Auswahl des Sensors, die zeitliche Eingrenzung, der Stil der Hintergrundkarte, die Farbskala sowie die Auflösung der hexagonalen Flächen angepasst werden. Durch diese freie Konfiguration sollen Lernende befähigt werden, unterschiedliche Darstellungsformen auszuprobieren, Zusammenhänge eigenständig zu erkennen und den Einfluss ihrer Darstellungsentscheidungen auf die Interpretation der Daten reflektieren zu können.

Theoretischer Bezug zu Data Literacy: Dieses Modul greift zentrale Aspekte der Förderung von Data Literacy auf, wie sie unter anderem von Wolff et al. (2016) und Ridsdale et al. (2015) beschrieben werden. Durch die eigenständige Auswahl von Darstellungsparametern wie Farbskalen, Kartentypen und Datenauflösung werden Lernende in die Lage versetzt, die Wirkung visueller Entscheidungen auf die Interpretation von Daten kritisch zu reflektieren. Wolff et al. (2016) betonen, dass die Arbeit mit unterschiedlichen Darstellungsformen und ästhetischen Elementen zur Entwicklung einer tieferen Datenkompetenz beiträgt, indem sie Analysefähigkeiten und kritisches Denken fördert. Darüber hinaus adressiert das Modul die Kompetenzbereiche *Interpretation von Daten* und *Evaluation von Darstellungen* nach Ridsdale et al. (2015), indem Lernende aktiv zwischen verschiedenen Repräsentationsformen wählen und deren Einfluss auf die Datenauswertung nachvollziehen können. Durch die Bewertung, ob eine Darstellung geeignet für die Präsentation von Daten ist, wird die Kompetenz *Präsentieren von Daten* gefördert.

Förderung von Teilkompetenzen aus dem Bereich Data Literacy nach Ridsdale et al. (2015): Durch die interaktive Auswahl und Anpassung von Darstellungsparametern werden insbesondere die Teilkompetenzen *Dateninterpretation* sowie *Evaluation und Sicherstellung der Qualität von Darstellungen* gefördert. Lernende werden dazu angeregt, die Auswirkungen von Visualisierungsentscheidungen auf die Wahrnehmung und Interpretation von Daten kritisch zu hinterfragen und verschiedene Darstellungsoptionen miteinander zu vergleichen. Zudem wird die Kompetenz *Explorative Datennutzung* unterstützt, da

durch die freie Gestaltung eigener Karten Hypothesen entwickelt und getestet werden können, was die eigenständige Analyse und Reflexion über Daten und deren Repräsentation stärkt.

Allgemeine Lernziele: Lernende sollen befähigt werden, eigene Visualisierungen auf Basis realer Sensordaten zu gestalten und die Wirkung unterschiedlicher Darstellungsoptionen auf die Interpretation von Daten kritisch zu reflektieren. Sie entwickeln ein Verständnis dafür, wie die Auswahl von Parametern wie Farbskala, Kartenstil oder Auflösung die Wahrnehmung von Mustern beeinflusst und lernen, eigene Analysewege im Umgang mit Umweltdaten zu entwickeln.

Mögliche User Story: Lernende erhalten die Aufgabe, ausgehend von einer eigenen Fragestellung ein Diagramm zu erstellen, das selbst gewählte Sensorwerte auf einer Karte darstellt. Sie experimentieren mit verschiedenen Farbskalen und Auflösungen, vergleichen die daraus resultierenden Darstellungen und diskutieren anschließend im Plenum, welche Einstellungen ihrer Meinung nach am besten geeignet sind, um bestimmte Trends oder Anomalien sichtbar zu machen.

3.2 Implementierung des DataLabs

Die datenverarbeitenden Schritte sind im wesentlich mit der Programmiersprache Python (Python Software Foundation, 2022) implementiert. Die Programmierung mit Python legte nah, ein ebenfalls auf dieser Programmiersprache aufbauendes geeignetes Webframework zu verwenden. Die Wahl fiel auf das Content-Management-System (CMS) Wagtail (Torchbox, 2024), das wiederum auf dem Webframework Django (Django Software Foundation, 2024) aufbaut.

Django setzt das Model-View-Controller (MVC) Prinzip um. Das bedeutet, dass es eine klare Trennung zwischen der Datenstruktur (dem Model), der Datenverarbeitung (dem Controller) und der Darstellung auf einer Webseite (dem View) gibt. Durch diesen modularen Ansatz gelingt es ein Projekt schnell und gut strukturiert um weitere Komponenten zu erweitern. Das CMS Wagtail erweitert Django um die Fähigkeit, dass auch Personen ohne Programmierkenntnisse die Webseite um Inhalte wie Texte und Bilder erweitern können. Die Ablage der Daten erfolgt in drei für ihre jeweiligen Zwecke geeigneten Datenbanken. Die grundsätzliche Konfiguration wird in der *objektrelationalen Datenbank* PostgreSQL (kurz Postgres) gespeichert.

Relationale Datenbanken (RDB) speichern Daten in Zeilen und Spalten, vergleichbar mit einfachen Tabellen. Es gibt eindeutige Spaltennamen und mittels eines Identifikators (ID oder auch Key) eindeutig beschriftete Zeilen, die in den jeweiligen Zellen die Werte enthalten. Sie sind deshalb relational, weil Zellen auf andere Tabellen verweisen können.

Ein geeignetes Beispiel für relationale Datenbankstrukturen ist die Speicherung von Sen-

sordaten in einem Messsystem. Jeder Sensor erhält dabei eine eindeutige Kennung, beispielsweise eine fortlaufende ID in einer separaten Sensorabelle. Zusätzlich werden die vom Sensor erfassten Messwerte – wie Temperatur, Luftfeuchtigkeit oder Windstärke – in einer weiteren Tabelle abgelegt. Zur strukturierten Verwaltung der Daten werden die Sensorinformationen (etwa Standort, Sensortyp) und die zugehörigen Messwerte voneinander getrennt gespeichert. Zwischen beiden Tabellen besteht eine Beziehung über die Sensor-ID. Erfasst ein Sensor regelmäßig Messwerte, handelt es sich um eine One-to-Many-Beziehung: Ein Sensor (ein Eintrag in der Sensorabelle) ist mit vielen Messwerten (mehrere Einträge in der Messwerttabelle) verknüpft. In manchen Szenarien ist es auch denkbar, dass mehrere Sensoren denselben Messwert aufzeichnen – etwa wenn verschiedene Sensoren redundant arbeiten oder auf denselben Datenpunkt referenzieren. In solchen Fällen kann eine Many-to-Many-Beziehung modelliert werden, bei der ein Messwert mehreren Sensoren zugeordnet ist und umgekehrt. Diese Struktur ermöglicht eine speicher-effiziente Ablage großer Datenmengen und eine performante Abfrage einzelner Sensoren oder Zeiträume in der Datenbank.

Objektelationale Datenbanken (ORDB) erweitern dieses Konzept, indem sich neben einfachen Datentypen wie Zahlen und Zeichenketten auch Objekte speichern und verknüpfen lassen. Ein Objekt in der Programmierung zeichnet sich unter anderem dadurch aus, dass es mehrere Daten zusammenführt, die zusammen gehören. An anderer Stelle wird erklärt wie eine SenseBox in JavaScript Object Notation (JSON) dargestellt wird (Abbildung 4). Neben dieser Darstellung für ein Objekt, lassen sich auch weitere Arten von Objekten in einer Datenbank wie Postgres speichern.

Für **Zeitreihen** (Time-Series) eignet sich das speziell für diesen Anwendungszweck entwickelte Datenbankmanagementsystem InfluxDB (kurz Influx). In manchen Anwendungsfällen ist es sinnvoller Daten in sogenannten Zeitreihen zu speichern. Das trifft zum Beispiel auf Daten zu, die einen zeitkritischen Aspekt haben, wie es bei Sensordaten oft der Fall ist. Üblicherweise werden Messwerte von vielen Sensoren abgerufen und gemeinsam in einer Datenbank gespeichert. Wenn die Messwerte in kurzen Intervallen (Millisekunden, Minuten, Stunden ...) übertragen werden, sammeln sich mit der Zeit sehr viele Daten an, was viel Speicherplatz belegt und durch die Informationsmenge die Geschwindigkeit reduziert mit der Informationen abgerufen werden können. Beispielsweise bei Temperaturdaten sind besonders aktuelle Werte interessant, weil sich danach Handlungen ableiten. Etwa wie hoch eine Heizung regelt, wie warm man sich anzieht, oder ob ein Gewächshaus belüftet werden muss. Umso länger die Daten in der Vergangenheit liegen, umso weniger ist die exakte zeitliche Auflösung von Interesse. Ab einem bestimmten Zeitpunkt ist nur noch der Mittelwert, von Minuten, Stunden und Tagen von Interesse. Das heißt es ist in dem Kontext sinnvoll die Mittelwerte zu berechnen und alle übrigen Werte aus der Datenbank zu entfernen. Je nach Anwendungszweck kann auch entschieden werden, dass nach einer Zeitspanne alle Daten wieder gelöscht werden.

Das bringt die Vorteile mit, dass sich Daten schneller auffinden lassen (es ist schneller nach dem gespeicherten Mittelwert für einen Tag zu suchen, als ihn bei jeder Abfrage neu zu berechnen) und spart den Speicherplatz der dafür nötig wäre. Ein weiterer Vorteil ist, dass bei dieser Art der Speicherung der benötigte Speicherplatz nach oben begrenzt wird. Der Speicherbedarf wächst nur bis zu einer definierten Größe an, die Datenbank wird anschließend um unnötige Daten befreit. Time-Series Datenbanken brauchen wie andere Datenbanken auch, einen eindeutigen Identifikator, um Daten verwechslungsfrei aus den Tabellen lesen zu können. Das was bei einer relationalen Datenbank durch die eindeutige ID jeder Zeile realisiert wird, wird bei Influx durch die Kombination aus einem Zeitwert (`_time`) und einer Kennung für den Sensor (`_measurement`) erreicht. Beide Werte zusammen sollten eindeutig sein und nur einmal in einer Zeile innerhalb einer Tabelle vorkommen. In dieser Art der Datenbank gibt es keine verknüpften Informationen, wo der Datensatz zu einem Objekt über mehrere Tabellen verteilt gespeichert wird.

Die dritte Datenbank ist eine *Schlüssel-Werte-Datenbank* (Key-value database) mit dem Namen Redis. Die Datenbank speichert die Tabellen ausschließlich im Hauptspeicher. Sie ist damit zwar deutlich schneller, als eine Datenbank die von der Festplatte gelesen wird, sie ist aber auch flüchtig und geht üblicherweise nach einem Neustart verloren. Die Daten werden nicht in einer Tabelle mit vielen Spalten gespeichert, wie in einer relationalen Datenbank, sondern mit einfach Schlüsseln (Keys) und Werten (Values). Jedem Key ist ein Value zugeordnet. Die Keys sind eindeutig. Wird ein Key ein zweites Mal in die Tabelle geschrieben, dann wird der vorherige Value gelöscht und mit dem neuen Wert überschrieben. Bei einer relationalen Datenbank ist das Überschreiben des Index-Werts nicht möglich und würde in der Standardkonfiguration einen Fehler provozieren. Diese Art von Datenbanken ist durch den einfachen Aufbau nicht zur Verwaltung von komplex verknüpften Informationen geeignet. Ihr Vorteil liegt in der einfachen Struktur und der hohen Geschwindigkeit, mit der sie auf die gespeicherten Daten zugreifen kann. Sie eignet sich damit als Cache für die Webseite. Wenn rechenintensive Operationen ausgeführt werden, kann das Ergebnis davon für einige Zeit im Hauptspeicher vorgehalten werden. Das erhöht die Geschwindigkeit der Webseite, insbesondere, wenn mehrere Personen innerhalb kurzer Zeit die gleichen Elemente aufrufen, und senkt zudem den Anspruch an die benötigte Rechenleistung.

3.2.1 Technologie Stack

Der Technologie Stack beschreibt, welche Techniken bei der Implementierung zum Einsatz gekommen sind (siehe Tabelle 4) und wie diese zueinander stehen (siehe Abbildung 2). Von außen nach innen gehend sind die Abstraktionsschichten, die sogenannten Layer abgebildet. Der äußere Layer, die VM (Virtual Machine), zählt zusammen mit den Containern (dick umrandet) zu der Systemschicht, sie bilden die Umgebung, in der die Anwendung läuft. Das Frontend (die Webseite oder allgemein die Benutzeroberfläche) und das Backend (die Logik und datenverarbeitende Teile) sind Bestandteile von Django. Ein Teil

des Backends erstreckt sich auf den Abschnitt Celery, eine asynchrone Task Queue. Die Datenbanken auf der linken Seite, PostgreSQL, InfluxDB und Redis, bilden die Persistenzschicht. Der Reverse-Proxy Caddy ermöglicht die Kommunikation zwischen Webbrower und der Anwendung. Die Pfeile markieren die Kommunikationswege.

Die Grundlage der Anwendung bildet ein POSIX-kompatibles System (Portable Operating System Interface), was auf alle unixoiden Systeme wie Linux, MacOS, BSD, ... zu trifft. Das System sollte in der Lage sein Python, sowie einige externe Programme wie cron, rsync, CMake sowie noch einige andere Werkzeuge (für Auflistung siehe Dockerfile in Tooling/prod in Abbildung 3) auszuführen. Die Umgebung wird durch Container realisiert, die alle Bestandteile vom Betriebssystem bis zum Programmcode des DataLabs enthält. Diese Container können gleichermaßen durch Docker oder dem dazu binärkompatiblen Podman bereitgestellt werden. Die Container wiederum laufen auf einem Host-System (Debian) in einer VM, das in der Lage ist die Container auszuführen. Innerhalb des Bereichs Container befinden sich die einzelnen Container Webserver, datalab und die Datenbanken. Container werden von ihrem Host-Sytem isoliert ausgeführt. Das bedeutet, ohne Freigabe sind die Programme innerhalb der Container zu keiner Kommunikation mit dem Host-System in der Lage. In dieser Anwendung wurden nur einige, durch Pfeile markierte, Kommunikationswege geöffnet. So ist die Kommunikation der Container untereinander möglich, es kann aber zum Beispiel nicht von einem Browser aus direkt auf die Datenbanken zugegriffen werden.

Tabelle 4: Tools des Technologischen Stacks und dessen jeweilige Beschreibung und Funktion.

Technologie	Beschreibung	Funktion im Projektkontext
Docker	Bereitstellung und Verwaltung von Anwendungen in isolierten Umgebungen	Ermöglicht eine reproduzierbare, portable und wartbare Infrastruktur für die gesamte Webanwendung
Caddy	Reverse-Proxy mit integrierter HTTPS-Unterstützung und einfacher Konfiguration	Webserver für die Webanwendung mit automatischer TLS-Konfiguration und effizientem Routing
PostgreSQL	Relationale Open-Source-Datenbank mit hoher Stabilität	Speicherung strukturierter Daten wie Inhalten, Benutzerdaten und Konfigurationen des Content-Management-Systems
InfluxDB	Zeitreihendatenbank zur effizienten Verarbeitung und Abfrage zeitlich indizierter Daten	Speicherung von Umweltdaten zur späteren Visualisierung
Redis	In-Memory-Datenspeicher mit extrem niedriger Latenz	Einsatz als Cache-Backend zur Beschleunigung von Datenbankabfragen und als temporärer Speicher für die Task-Queue
Celery	Verteiltes Task-Queue-System zur asynchronen Ausführung von Hintergrundaufgaben	Verarbeitung zeitintensiver oder regelmäßig geplanter Aufgaben wie Datenimporte außerhalb des Request-Response-Zyklus
Django / Wagtail	Django: Webframework für Python; Wagtail: darauf aufbauendes CMS	Basis der Webapplikation für Backend-Logik, API-Anbindung und redaktionelle Inhaltsverwaltung
htmx	JavaScript-Bibliothek zur Umsetzung dynamischer Webinteraktionen mittels HTML-Attributen	Ermöglicht asynchrone Inhaltsaktualisierungen (z. B. Diagramme, Filter) ohne vollständiges Neuladen der Seite
Plotly	Framework zur Erstellung interaktiver, webbasierten Diagramme mit Fokus auf Datenexploration	Darstellung komplexer Umweltdaten zur Förderung von Data Literacy durch interaktive Visualisierungen
Leaflet	JavaScript-Bibliothek zur Einbettung interaktiver Karten in Webseiten	Geovisualisierung von standortbezogenen Datenquellen
Bootstrap	CSS-Framework für responsives und modulares Webdesign	Gestaltung einer adaptiven Benutzeroberfläche mit konsistentem Layout und mobilen Ansichten

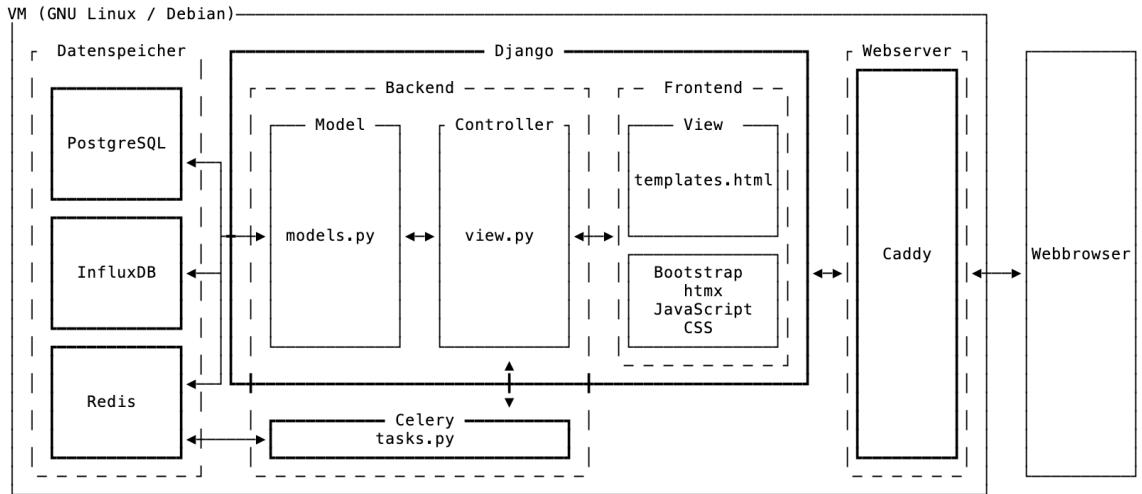


Abbildung 2: Der Softwarestack des DataLabs.

3.2.2 Struktur des DataLabs

Wie eingangs erwähnt wurde, setzt diese Arbeit auf der Idee des Model-View-Controller (MVC) Prinzips auf. Das Model beschreibt, wie die Daten in der Datenbank abgebildet werden sollen und wie die Verbindung zwischen den einzelnen Daten hergestellt wird. Das Model beschreibt also ganz konkret, wie die abgerufenen Daten der SenseBoxen in die Datenbank geschrieben werden sollen. Ein Model besteht dazu unter anderem aus dem Namen, was den Tabellennamen in der Datenbank entspricht, sowie den Fields, was den Spalten in der Tabelle entspricht. Der Eintrag `class Person(models.Model)` entspricht einer Tabelle mit der Bezeichnung `Person` in der Datenbank. Das Field `name = models.CharField()` entspricht dann der Spalte `name` in der Tabelle `Person`. Die Spalte darf nach dieser Festlegung nur Textzeichen (char) enthalten. Alle Models werden in Dateien mit der Bezeichnung `models.py` abgelegt (siehe Abbildung 3).

Das Wurzelverzeichnis (root, im Kasten), trägt den Namen des Projekts. Unterhalb von root befinden sich die Apps (logische Einheiten innerhalb von Django) `datalab`, `home` und `core`. Die erste App nach Erstellung des Projekts muss nach Konvention den gleich Namen tragen wie das Projekt selbst. Innerhalb der App `datalab` werden grundsätzliche Einstellungen (`settings`) getätigt und grundlegende Ressourcen für das Frontend verwaltet (`static` und `templates`). Die App `home` steht für alle Funktionen, die direkt nach Aufruf der Webseite dargestellt werden können. Die Dateien `models.py`, `views.py` und `templates` folgend dem MVC Schema. Die Datei `urls.py` definiert die URLs, über die Funktionen innerhalb von `views.py` aufgerufen werden können. In der App `core` werden Kernfunktionen für das DataLab beschrieben. Innerhalb von `management` befinden sich Funktionen, um für den Betrieb wichtige Aufgaben durchzuführen (siehe Abschnitt 3.3 Abruf von Umweltdaten). In `tools.py` werden Funktionen zentral beschrieben, die an mehreren Stellen des Programms benötigt werden. `tasks.py` enthält die Bestandteile, die innerhalb der asynchronen Task Queue Celery ausgeführt werden. Unter

den Apps werden einige weitere Funktionen näher ausgeführt: In dem Ordner `locale` befinden sich Textdateien mit der Endung `.po`, welche genutzt werden können, um die Webseite in die dort erwähnten Sprachen zu übersetzen. In dem Ordner `Tooling` wird definiert, wie sich die Entwicklungs- (`dev`) und die Produktivumgebung (`prod`) bauen lassen. Die Dateien mit `compose.yml` im Namen beschreiben, wie die Container erstellt werden. In `cron` werden die `management-commands` nach festen Zeitintervallen aufgerufen. Die Datei `entrypoint.sh` definiert kontextabhängig die Startbedingungen. Die Dateien mit `.env` im Namen enthalten lokale Variablen, Passwörter und Schlüssel, die den Zugang zu internen und externen Ressourcen ermöglichen. Und die Datei `uploads.sh` enthält ein Skript, womit das Programm auf einem Server deployed werden kann.

Die Bezeichnung *View* ist in Django leider etwas irreführend. Entgegen der ersten Annahme handelt es sich hier nicht um die Darstellung von Informationen, sondern um den Teil, in dem im wesentlichen Berechnungen stattfinden. Die Views bestehen aus Python Funktionen (in manchen, aber hier nicht ausgeführten Fällen auch aus Klassen (`classes`)), die durch eine URL aufgerufen werden können. Dem Funktionsaufruf können Argumente übergeben werden, wodurch das Verhalten der Funktionen gesteuert werden kann. Die Funktionen geben üblicherweise eine Antwort (`return Wert`), die in Form einer Webseite gerendert werden kann. Die Views werden in Dateien mit der Bezeichnung `views.py` abgelegt. Model und View gehören zum Backend des Webservers. Der *Controller* (hier: Template) ist genauso irreführend in Django bezeichnet und beschreibt alles, was für die Darstellung von Informationen gedacht ist. Hier wird also definiert, was und wie auf der Webseite dargestellt werden soll. Dazu gehören alle Dateien innerhalb der Ordner `templates` und `static` und bilden das Frontend. Sie sind im Gegensatz zu dem Backend nicht aus Python-Code, sondern aus HTML, CSS, JavaScript (`htmx`) und, in diesem Fall, aus der Django Template Language (Django Software Foundation, 2025) aufgebaut. Die Django Template Language kann Python Typen und Objekte verarbeiten, wodurch ein kleiner Teil der Logik in das Frontend verlagert werden kann.

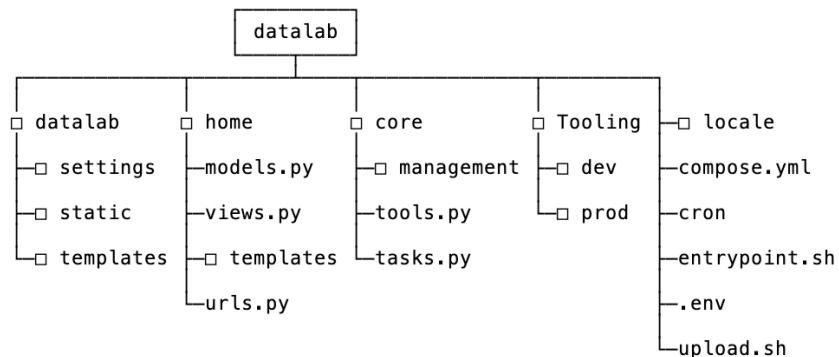


Abbildung 3: Darstellung der wesentlichen Ordnerstruktur des DataLabs.

3.3 Abruf der API

Das DataLab soll aktuelle und aufgezeichnete Daten aus unterschiedlichen Quellen in Grafiken darstellen können. Eine dieser Datenquellen sind SenseBoxen. SenseBoxen sind eine Art Plattform, um Umweltsensoren im Internet verfügbar zu machen. Die Daten der SenseBoxen können auf zwei Wegen abgerufen werden. Einmal über eine Webseite, die openSenseMap („openSenseMap.org“, 2025), die alle verfügbaren SenseBoxen auf einer Landkarte verzeichnet. Bei Auswahl einer SenseBox auf der Webseite werden Sensordaten in einfachen Diagrammen dargestellt.

Der zweite Weg auf die Messdaten zuzugreifen ist über die openSenseMap-API (Reedu GmbH & Co. KG, 2025). Eine API (application programming interface = Programmierschnittstelle) beschreibt eine Methode, um programmatisch über Textdateien mit einer Anwendung zu kommunizieren. Zunächst muss eine passende API-Anfrage gestaltet werden, wie sie in der API-Dokumentation beschrieben wird. Eine Anfrage an die URL

<https://api.opensensemap.org/boxes/5bf93be1a8af82001afc1568>

wird von der openSenseMap-API in Form von maschinenlesbaren Text beantwortet, wie es ist der Abbildung 4 dargestellt wird. Auf diese Weise ist es möglich ohne Umwege über eine grafische Oberfläche direkt mit der Datenbank der SenseMap zu kommunizieren. Die für das DataLab relevanten Informationen werden nun aus dem Text herausgelesen und in die lokale Datenbank geschrieben. Dazu wird die API ORM (Object-relational mapping) von Django verwendet. Das ORM beschreibt wieder eine Methode, um beliebige Informationen (einfache Datentypen sowie Objekte) in die Datenbank zu schreiben und von dort zu lesen.

Die Funktion die für das Abrufen der openSenseMap-API und Speichern der Werte zuständig ist, gehört nicht zum MVC-Schema, da dieses nur ausgeführt wird, wenn die Webseite, die durch das DataLab bereit gestellt wird, aufgerufen wird (siehe Abschnitt 3.2.2 Struktur des DataLabs, wie ein View ausgeführt wird). Die Messwerte der SenseBoxen sollen in festen Zeitintervallen von der API geholt werden. Das wird über zwei Tools umgesetzt. Das erste Tool ist ein management-command. Das sind spezielle Funktionen neben dem MVC Schema, die dafür geeignet sind im Hintergrund Arbeiten an der Datenbank auszuführen. Das management-command `collect_data.py` wurde gebaut, um Daten über die openSenseMap-API abzurufen und die Antworten zu speichern. Für die regelmäßige Ausführung dieser Funktion sorgt das Programm cron. Es ruft jede volle Stunde das Command auf (siehe Abschnitt 3.2.2 Struktur des DataLabs, cron).

Aufbau der Funktion `collect_data.py`

1. API Abrufen: Tabelle aller SenseBoxen mit

`get_latest_boxes_with_distance_as_df()`: Fragt die API nach allen SenseBoxen in einer bestimmten Region. -> Die Antwort ist eine Tabelle mit IDs, Name,

letzter Meldung, verfügbaren Sensoren und Standort.

2. **API Abrufen: Tabelle aller Sensor-Messwerte mit**

`get_sensebox_data()`: Es werden alle Sensoren, von allen in der Tabelle verfügbaren SenseBoxen, abgefragt. -> Die Antwort ist eine Tabelle mit ID, Sensorname, Zeit, Messwert und hinterlegte Einheit für den Messwert.

3. **Schreiben der Messwerte in die Datenbank mit**

`write_to_influx()`: Alle Messwerte werden in die Time-Series Datenbank Influx geschrieben.

4. **Cache neu generieren mit `cache.set()`**: Zeitkritische und rechenintensive Operation werden ausgeführt, um das Ergebnis davon in den Cache (Redis Datenbank) zu speichern.

3.3.1 Tabelle aller SenseBoxen

`get_latest_boxes_with_distance_as_df()`

Das Ziel dieser Funktion ist es Werte aller SenseBoxen in den gewünschten Regionen abzurufen und als strukturierte Tabelle für die weitere Verwendung zu speichern. Die openSenseMap-API antwortet mit einem Datensatz für jede einzelne SenseBox unter anderem bestehend aus einer eindeutigen ID (`_id`), einem frei gewählten Namen (`name`), dem genauen Ort (`currentLocation`), einer Auflistung aller Sensoren (`sensors`), die Bestandteil der jeweiligen SenseBox sind, sowie dem Datum der letzten erfolgreich übertragenen Messung (`lastMeasurementAt`). Der Datensatz enthält noch weitere Werte, die an anderer Stelle relevant werden.

Die Abbildung 4 enthält einen Auszug eines exemplarischen Datensatzes im JavaScript Object Notation (JSON) Format, in der Form wie er von der openSenseMap-API zurück gegeben wird. Bei JSON handelt es sich um einen de-fakto-Standard für Datenaustausch im Internet für maschinenlesbaren Text. Objekte sind in geschweiften Klammern (`{...}`) eingeschlossen, Elemente beliebigen Typs befinden sich in Listen (synonym Arrays), welche mit eckigen Klammern (`[...]`) notiert werden. Die Elemente dieses JSON sind fiktiv und dienen nur der Veranschaulichung. Auslassungspunkte zeigen, dass hier beliebig viele weitere Elemente des gleichen Typs folgen können.

Aufbau der Funktion:

1. Abfrage der Funktionsparameter
2. Abfrage des Cache
3. Abruf der API um eine Liste von allen SenseBox IDs der gewählten Regionen in Form eines pandas DataFrames (`df`) zu erhalten

```
[
  {
    "_id": "5bf93be1a8af82001afc1568",
    "model": "luftdaten_sds011_dht22",
    "name": "Victoriastadt",
    "currentLocation": {
      "type": "Point",
      "coordinates": [
        13.479055,
        52.502998,
        35
      ],
      "timestamp": "2024-03-06T21:04:09.174Z"
    },
    "exposure": "outdoor",
    "sensors": [
      {
        "_id": "65e8da49cbf5700007fb11a6",
        "title": "PM10",
        "unit": "µg/m³",
        "sensorType": "SDS_011",
        "icon": "osem-cloud",
        "lastMeasurement": "65f569298c845600074a3eb3"
      },
      ...
    ],
    "model": "custom",
    "groupTag": [
      "Futurium",
      "Luftdaten"
    ],
    "lastMeasurementAt": "2022-12-14T13:41:29.298Z"
  },
  ...
]
```

Abbildung 4: Auszug einer in JSON formatierten Antwort der openSenseMap-API.

- (a) Abruf der Field-Werte von SenseBoxLocation
 - (b) Konstruktion eines Dictionarys (`params`) zum Abruf der API
 - (c) Konstruktion der URL aus `params`
 - (d) Abruf der API, Ergebnis davon ist ein json (`r_json`)
 - (e) einlesen des json in ein df (`temp_df`)
 - (f) den df zur Liste aller df (`frames`) hinzufügen
 - (g) die Liste aller df zu einem df konkatenieren
4. Bereinigung des df
 - (a) interpretieren des Values `lastMeasurementAt` als Zeit
 - (b) fehlerhafte Zeitwerte entfernen
 - (c) Zeitwerte auf volle Minuten runden
 - (d) nur Zeitwerte vom letzten aktuellen Datum behalten
 5. df in den Cache schreiben
 6. df zurückgeben

Wie die meisten Funktionen dieses Programms ist auch diese asynchron geschrieben, um es zu ermöglichen, mehrere Aufgaben parallel auszuführen, ohne dass diese das Programm

blockieren können. Einige Bestandteile werden an die asynchrone Task Queue Celery übergeben. Durch die Übergabe können die Prozesse parallel zueinander ausgeführt werden. Diese Funktion nimmt die optionalen Parameter `region` und `cache_time` entgegen und gibt einen Pandas DataFrame zurück.

Ein Pandas (pandas development team, Wes McKinney, 2024) DataFrame (`df`) ist ein mit einer Tabelle vergleichbares Datenformat für die Programmiersprache Python, was geeignet ist Daten zu verarbeiten und zu analysieren. Der Pandas DataFrame ist in dieser Arbeit ein zentrales Werkzeug zur Verarbeitung von Daten.

Der Parameter `region` bezieht sich auf ein Objekt des Models `SenseBoxLocation` und definiert einen Bereich mit frei zu vergebenen Namen und einigen dafür relevanten Feldern (siehe Model `SenseBoxLocation`). Der Standardwert (default) wird hier mit `all` angegeben, was eine Umschreibung für alle Regionen ist. Der Wert `Berlin` würde alle SenseBox-IDs der Region Berlin abrufen. Der Parameter `cache_time` erhält den default Wert `60`, was die Zeit in Sekunden vorgibt, die der Cache gültig sein soll. Die Werte beider Parameter können bei Funktionsaufruf über das keyword-Argument neu belegt werden.

Die Funktion `get_latest_boxes_with_distance_as_df()` prüft zunächst, ob es einen bestehenden Eintrag im Cache mit einem regionsspezifischen Key gibt. Wenn nicht dann werden die relevanten Daten durch API-Abfragen gesammelt, im Cache gespeichert und als `df` zurückgegeben. Wenn ein Eintrag im Cache gefunden wurde, wird er ohne weitere Berechnung direkt als `df` zurückgegeben. In dem Fall, dass kein Eintrag im Cache gefunden wurde, wird die asynchrone Funktion `create_df()` definiert. Sie nimmt die internen Argumente `_frames` (eine zunächst leere Liste von `df`) und `_location` (steht für einen konkreten Eintrag aus der Tabelle `SenseBoxLocation`) entgegen.

Die Funktion entnimmt die Werte aller hier relevanten Fields aus dem `_location` Eintrag (`location_longitude`, `location_latitude`, `maxDistance` und `exposure`) erzeugt ein Dictionary (ein spezielles Python Objekt zur Speicherung von Key-Value-Paaren, kurz dict), welches die Parameter enthält, um die URL über die asynchrone Funktion `get_boxes_with_distance()` zu erzeugen. Die Funktion zur URL-Erzeugung ruft überdies auch die asynchrone Funktion `get_url_async()` auf, die einen generalisierten Ansatz implementiert, um effizient eine große Menge an URLs parallel abzurufen und dabei auf diverse eventuelle Netzwerkprobleme angemessen reagieren zu können. Das Ergebnis davon ist ein JavaScript-Objekt (`json`) mit dem Namen `r_json`. Das `json` wird interpretiert und in ein Pandas `df` überführt, was letztlich als `_frames` zurück gegeben wird. Jedes auf diese Weise erzeugte `df` wird zu der Liste `frames` hinzugefügt. Jedes `df` entspricht den gesammelten IDs einer Region. Im letzten Schritt dieses Abschnitts werden die Daten aller Regionen konkateniert, was in diesem Fall in einem einzigen `df` resultiert.

Das `df` erfährt noch einige Anpassungen, damit die Werte kompatibel mit der Time-Series Datenbank InfluxDB (Influx) wird. Influx erwartet als Index einen Zeitwert. Um diesen zu

erzeugen wird versucht die noch als String hinterlegten Werte in der Spalte `lastMeasurementAt` als ein Objekt der Pandas Klasse `DateTime` zu interpretieren. In den Fällen wo das nicht funktioniert, meist weil die Werte schlicht fehlen, oder weil es eine nicht zu interpretierende Zeichenfolge gab, werden die kompletten Zeilen aus dem df entfernt. Anschließend werden die Zeitwerte, die ursprünglich eine Genauigkeit im Bereich von Nanosekunden hatten, auf volle Minuten abgerundet. Das macht die Daten vergleichbar, da sie ab jetzt immer vollen Minutenwerten entsprechen. Zusätzlich findet durch diese Konvertierung eine Bereinigung statt. Dadurch, dass Influx die Eindeutigkeit der Werte über den Zeitstempel (`time`) und eine Bezeichnung für die Datenreihe (`_measurement`) herstellt, kann es keine zwei Einträge mit exakt gleichen Werten für `_time` und `_measurement` in einer Tabelle geben. Der Wert `_measurement` dient in Influx als eindeutige Kennung, ähnlich einer ID. Da jede SenseBox ebenfalls eine eindeutige Kennung trägt (Dopplungen also ausgeschlossen sind) wurde die ID als Wert hierfür verwendet. Bei dem Import in Influx wird somit maximal ein Wert pro Minute für jede Datenreihe (die für eine SenseBox steht) hinterlegt, was den Speicherbedarf reduziert.

Das Design der Abfrage an die API ist so gestaltet, dass sie nach Möglichkeit schnell antwortet. Die Antwortzeiten oder das Design einer geeigneten Abfrage ist nicht in der Dokumentation enthalten. Die schnellsten Antwortzeiten der API wurden daher durch eine Testreihe ermittelt, die alle Möglichkeitskombinationen der Parameter durchprobierte. Die API-Abfrage ist dadurch sehr schnell, enthält dann aber auch IDs von inaktiven SenseBoxen, die seit längerer Zeit keine Werte übermittelt haben. Um diese inaktiven SenseBoxen aus den Ergebnissen herauszufiltern werden alle Zeilen aus dem df entfernt, die ältere Werte als vom aktuellen Tag enthalten. Der auf diese Weise erstellte und bereinigte df wird in den Cache geschrieben und als Funktionswert zurückgegeben.

3.3.2 Zeitfenster mit einem Delta berechnen

`get_timeframe()`

Das Ziel der Funktion `get_timeframe()` soll es sein, die Messwerte aller Sensoren von allen in der Datenbank hinterlegten SenseBoxen abzuholen. Ein API-Abruf bestehend aus der SenseBox-ID, sowie der Sensor-ID enthält die letzten 10.000 Messwerte jedes einzelnen Sensors. Zum Zeitpunkt des Verfassens dieser Arbeit versenden in Berlin über 112 SenseBoxen aktuelle Daten (nur aktive Boxen werden vom Programm auch berücksichtigt). An diesen SenseBoxen sind insgesamt 419 Sensoren angeschlossen. Das entspricht über 4 Millionen Werten, die bei jedem Abruf übermittelt werden würden. Das DataLab kann diese Menge an Daten innerhalb von etwa 7 Sekunden in die Datenbank schreiben. Das Abrufen hingegen dauert auf diese Weise etwa 6 Minuten, wenn die API die Anfrage durch Überlastung nicht vorher abbricht. Die API erlaubt es den Umfang der Daten einzuschränken, indem vorgegeben wird, in welchem Zeitfenster (`timeframe`) die Daten abgerufen werden sollen. Die Funktion `get_timeframe()` nimmt den Parameter `time_delta`

entgegen, womit das Zeitfenster in der Länge definiert werden kann. Es wird zunächst die aktuelle Zeit in einem Python datetime-Objekt ermittelt, dann wird über das Delta ein, in der Vergangenheit liegender, Zeitpunkt bestimmt und die beiden Datumswerte in die militärische Zeitzone Zulu konvertiert, dem Format, das die API als gültiges Zeitformat akzeptiert. Eine Abfrage enthält damit nur Werte des aktuellen Tages, sofern `time_delta` nicht anders vorgegeben wurde. Die Funktion `collect_data.py` benötigt damit für einen kompletten Durchlauf etwa 20 bis 30 Sekunden, statt mehreren Minuten und verhindert, dass die API die Anzahl der Anfragen nicht mehr beantworten kann und die Antwort abbricht. Dieses Vorgehen schont die Ressourcen der API, sorgt für stets aktuelle Daten und beschleunigt den Abruf, was wiederum den Nutzerinnen und Nutzern des DataLabs zu Gute kommt.

Es gibt einen Feature Request („Migrate to Postgres · Issue #846“, 2023) für die openSenseMap-API, der einen Wechsel zu Postgres vorschlägt, was die Datenbankabfrage erheblich robuster und schneller machen würde, sowie den Vorschlag in Echtzeit Daten abrufen zu können („Real time · Issue #40“, 2016). Beides würde die Komplexität von API-Abfragen erheblich reduzieren, die Geschwindigkeit erhöhen und die Wahrscheinlichkeit für Fehler verringern.

3.3.3 Messwerte der SenseBoxen abrufen

`get_sensebox_data()`

Das Ziel dieser Funktion ist es die Sensorwerte aller SenseBoxen abzurufen, die in der Tabelle von `get_latest_boxes_with_distance_as_df()` gespeichert wurden, und sie in Influx zu übertragen. Zunächst wird die Tabelle aller SenseBoxen zusammen mit dem Ergebnis von `get_timeframe()` an die Funktion `run_multithreaded()` übergeben. Dort wird die Tabelle zunächst zeilenweise in sogenannte Series zerlegt. Eine Zeile entspricht immer einer SenseBox. Die Series werden anschließend zusammen mit dem Timeframe als Aufgaben (Tasks) an die Funktion `get_sensebox_data()` übergeben. Die Bearbeitung erfolgt, wie der Name der umschließenden Funktion schon andeutet, multi-threaded. Das bedeutet, dass zeitintensive Prozesse, wie hier die Abfrage an eine API, parallel erfolgen können und sich nicht gegenseitig blockieren, indem sie serialisiert ausgeführt werden. Das erhöht die Ausführungsgeschwindigkeit um ein Vielfaches und wird hauptsächlich durch die Antwortgeschwindigkeit der externen API begrenzt. Die Messwerte werden ohne eine Einheit übertragen und die Bezeichnung des Sensors ist nicht immer einheitlich. Die Funktion `get_sensebox_data()` sorgt für eine einheitliche Bezeichnung der Sensoren (zum Beispiel aus „Temperature“, „Lufttemperatur“, „Temperatur (DHT11)“ und „temperature“ wird einheitlich „Temperatur“) und ergänzt die fehlenden Einheiten. Sie extrahiert auch die GroupTags aus den Beschreibungen, da diese für einige der später gewählten Module notwendig sind. Lernende sollen später in der Lage sein ihre individuell benannte SenseBox in der großen Auswahl an verfügbaren SenseBoxen

schnell und intuitiv finden zu können. Unter anderem in dem DataLab Modul *Live vor Ort - Sensordaten am Standort* (Abschnitt 3.4.4) erfolgt die Auswahl der SenseBoxen anhand der GroupTags. Die Messwerte kommen mit einem Zeitpunkt der Messung, der allerdings nur als eine Zeichenkette und in der militärischen Zeitzone Zulu formatiert ist. Die Zeichenkette wird zunächst in einem, zu Python kompatiblen Zeitformat neu interpretiert und auf Minuten gerundet abgespeichert. Die minutengenaue Speicherung der Daten ist genau genug für das DataLab als ein Werkzeug zur Vermittlung von Data Literacy und rundet die Werte zusätzlich sinnvoll, damit sie untereinander vergleichbar werden. Einige der Daten werden an dieser Stelle in die Datenbank Postgres zur späteren Verwendung gespeichert. Dazu gehören die SenseBox IDs, die dazu gehörigen Namen, die Geokoordinaten, die GroupTags, sowie die Einheiten der Messwerte. Alle anderen Werte werden als eine Liste von Tabellen in Form eines df an die Funktion `collect_data.py` zurück gegeben.

3.3.4 Messwerte in die Datenbank schreiben

`write_to_influx()`

Die Funktion `write_to_influx()` wird über eine Schleife aufgerufen, wobei ihr bei jedem Durchlauf eine Tabelle mit Sensorwerten übergeben wird. Die Funktion stellt darauf hin eine Verbindung zu der Influx-API her und übergibt ihr die Tabelle an Messwerten. Dankenswerterweise akzeptiert Influx dabei den df als Eingabe, so dass keine weiteren Bearbeitungsschritte notwendig sind. Influx ist bei der Verarbeitung auf einen gültigen Zeitstempel, sowie einem Bezeichner für den Messwert angewiesen. Beide wurden in der vorherigen Funktion erzeugt. Es wird die Bezeichnung des Spaltenüberschrift verwendet, also in diesem Fall der Name des Sensors. Influx erzeugt selbstständig über den Bezeichner `data_frame_measurement_name` eine passende Zuweisung mittels der SenseBox ID. Die Daten werden damit strukturiert abgelegt, es ist nicht notwendig die Tabelle oder in ihr enthaltende Spalten manuell anzulegen.

3.4 Implementierung der DataLab Module

3.4.1 Messstellen-Karte

Das Ziel dieser Darstellung besteht darin, dass sich die Lernenden dem Konzept Umweltdaten nähern und über die Darstellung der Standorte auf einer Landkarte ein erstes Verständnis dafür aufzubauen, an welchen Orten Daten gesammelt werden. Die Messdaten werden in einfachen Achsendiagrammen angezeigt, was grundlegende Kompetenzen zum Verstehen von Daten in Form von Diagrammen fördern kann, siehe Ridsdale et al. (2015), *Conceptional Framework, Introduction to Data*.

Zur Darstellung der Diagramme wird der View `draw_graph(sensebox_id)` aufgerufen. Neben dem aktuellen Kontext aus dem Abfrage (request), wird auch die senseBox

ID (`sensebox_id`) übertragen. Auf Grundlage dieser ID wird die InfluxDB nach Werten dieser Box der vergangenen 72 Stunden gefragt. Die Antwort wird in ein df überführt. Die Spaltennamen im df enthalten die Bezeichnungen für die Sensoren. Die Bezeichnungen werden verwendet, um die dazu passenden Einheiten aus der Tabelle `SensorsInfoTable` abzufragen und zusammen mit der Bezeichnung in die Spaltennamen zu schreiben. Die Tabelle mit den Einheiten wurde im Rahmen den Aufrufs von `collect_data.py` mit Werten gefüllt und teilweise automatisch und später händisch angepasst, um ein einheitlich korrektes Format für die Einheiten zu erhalten.

Die Tabelle enthält zeilenweise die Uhrzeiten und mit den dazu passenden Messwerten aller Sensoren dieser SenseBox. Um die Menge an Daten zu reduzieren, die später auf das Endgerät übertragen werden soll, wird der Median über 5-Minuten-Intervalle berechnet und in die Tabelle gespeichert. Alle übrigen Zeitpunkte werden aus der Tabelle entfernt. Das hat zusätzlich den Effekt, dass Zeitpunkte auch zwischen den Diagrammen vergleichbar werden, da die Messwerte immer in diesen Intervallen dargestellt werden.

Anschließend wird eine Berechnung durchgeführt, die die einheitliche Darstellung der Diagramme sicherstellt. Dazu wird die Anzahl der Spalten mit Messwerten bestimmt. Danach wird eine minimale Höhe für alle Diagramme festgelegt. Danach wird ermittelt, was die längste Bezeichnung einer Spalte aus der Reihe der Messwerte mit der dazu passenden Einheit ist. Eine Formel berechnet, wie hoch die Diagramme mindestens sein müssen, damit die Bezeichnungen für die Ordinate nicht zu lang werden und über das Diagramm hinaus ragen. Es wird ermittelt, welcher der Werte (die minimale Höhe oder der eben berechnete Wert) größer ist und als Wert für den Höhe aller Diagramme festgelegt. Ohne diese Berechnung könnte es zu Überschneidungen der Bezeichnungen für die Ordinaten kommen.

Nachdem die Bearbeitung der Messwerttabelle abgeschlossen ist und sichergestellt wurde, dass die Diagramme später unter allen Umständen lesbar sein werden, werden die Diagramme mit `plotly` (Kruchten et al., 2024) gezeichnet. Dazu wird die Anzahl der Spalten mit Messwerten bestimmt. Es wird ein Diagramm erzeugt mit der dazu passenden Anzahl an Unterdiagrammen (sub plots). Jedes dieser Diagramme wird einen Zeitreihe eines Messwerts darstellen. Es gibt zwei Arten von sub plots, die auf dieser Basis gezeichnet werden können: Liniendiagramme für die Darstellung von geschlossenen Messreihen. Wenn die Messungen seltener als in 5-Minuten-Intervallen durchgeführt wurden, wird ein Streudiagramm gezeichnet. In diesem Fall werden die Messpunkte nicht mehr mit einer Linie verbunden. Es ist auch vorgesehen, dass beide Diagrammarten miteinander gemischt werden können, wenn es Messwerte aus beiden Kategorien gibt.

Zur Bestimmung, welche Art der Darstellung gewählt werden soll, werden die Stellen ermittelt, für die es keine Messwerte (`NaN - Not a Number`) gibt. Wenn dieser Messwert vorher und nachher ebenfalls von einem fehlenden Messwert flankiert wird, wird die Er-

kenntnis in einer Liste vermerkt. Diese Liste wird mit der Liste aller Messwerte verglichen. Für alle Zeitpunkte, für die ein Vermerkt hinterlegt wurde, wird in Anschluss der Messwert in Form eines Kreises im Diagramm verzeichnet (Streudiagramm). Alle anderen Werte werden mit einer Linie verbunden (Liniendiagramm), oder im Fall von einfachen Fehlstellen, also ohne flankierende fehlende Messwerte, wird die Stelle einfach nicht eingezeichnet. Es ist auch möglich, dass beide Formen der Darstellung auch gemischt in einem Diagramm auftreten, wenn beide Kriterien abwechselnd zutreffen. Diese Art der Behandlung ist notwendig, da manche SenseBoxen ihre Daten unregelmäßig übermitteln und eine Darstellung in diesen Fällen dann nicht mehr möglich wäre oder es bei dem Betrachter zu falschen Rückschlüssen führen könnte.

Nach der Entscheidung, welche Art von Diagramm gezeichnet werden soll, kommt eine Funktion, die gesetzliche Grenzwerte einzeichnen und bei Überschreitung Messwerte farblich hervorheben kann. Die Funktion `red_shape_creator(threshold, df, item, row)` nimmt dazu die Tabelle aller Messwerte (`df`), den Grenzwert (`threshold`), sowie den Namen der Spalte (`item`) entgegen. Der Parameter `row` spielt nur dafür eine Rolle, falls in Zukunft andere Arten von Diagrammen gezeichnet werden sollen. Die Funktion ermittelt zunächst, welche der ausgewählten Werte über dem Threshold liegen und vermerkt diese als `True` und `False` Werte in einer Liste. Die Wahrheitswerte werden in `0` und `1` konvertiert. Nur die Stellen in der Liste sind interessant, an denen ein Wertepaar den Threshold passiert. Die Übergänge in beide Richtungen werden notiert. Es gibt noch einige Konvertierungsschritte, die sicherstellen, dass die Darstellung auch für verschiedene Randbedingungen gilt. Das Ergebnis davon wird mit den Zeitwerten in der Tabelle abgeglichen. Es wird die Anweisung gegeben, dass überall dort, wo der Threshold passiert wurde eine rote Markierung beginnen beziehungsweise enden soll. Das Ergebnis davon ist in der Abbildung 6 zu sehen. Diese Funktion kann anschließend auf beliebige Diagramme angewandt werden, sofern es gewünscht ist, dort Grenzwerte darzustellen.

Ein zentraler Aspekt gilt für alle Kartendarstellungen im DataLab:

Unabhängig davon, ob es sich um vektorbasierte Inhalte wie 3D-Gebäude oder um Bilder von Straßenkarten und Satellitenaufnahmen handelt – sämtliche extern geladenen Karten-elemente werden nach dem ersten Aufruf im Cache des DataLabs zwischengespeichert. Dies hat mehrere Vorteile: Zum einen werden die Server der jeweiligen Anbieter entlastet, wodurch das Risiko einer Sperrung aufgrund massenhafter Anfragen vermieden wird. Zum anderen können die benötigten Kartenressourcen nach dem Erstabruf deutlich schneller an die Endgeräte der Lernenden ausgeliefert werden, was die Performance und Nutzungs-freundlichkeit in der Anwendung spürbar verbessert.

3.4.2 Hexagonale Dichtekarte

Dieser Diagrammtyp nutzt Hexagone (Zusammenfassung von Daten in sechseckigen Figuren = Hexagonal Binning), um beliebige von der openSenseMap-API übertragene Sen-

sorwerte auf einer Landkarte darzustellen. Das folgende Beispiel bezieht sich auf die Darstellung von Temperaturwerten. Wie im Abschnitt 3.4.6 ausgeführt wird, lassen sich die hier erwähnten Parameter anpassen.

Die Darstellung in Form von Sechsecken eignet sich, um eine große Anzahl von Daten auf einer Landkarte darzustellen, wie Carr et al. (1987) erstmals ausführen. Gleichseitige Sechsecke gehören zu den wenigen geometrischen Figuren, mit denen sich eine Ebene, ohne die Verwendung von weiteren Figuren, lückenlos belegen lässt. Sonst ist das nur mit Dreiecken, Vierecken und den kürzlich entdeckten Einstein-Kacheln (Smith et al., 2024) möglich.

Das gleichseitige Sechseck wurde der Darstellung von Messwerten auf einer Choroplethenkarte (choropleth map, auch Flächenkartogramm) bevorzugt. Bei dieser Kartendarstellung werden Gebiete nach dem Auftreten und der Ausprägung von Objekten eingefärbt (siehe dazu 4.3). Das kann dazu führen, dass Gebiete auf der Karte gar nicht eingefärbt werden, wenn es dafür keine Messwerte gibt, oder ein Gebiet ganzflächig eingefärbt wird, was dann den Eindruck erweckt, dass der Wert auch in Realität in dem gesamten Gebiet gemessen wurde. Flächenkartogramme werden auf der Grundlage von Geoinformationen gezeichnet, was bedeutet, dass es eine einheitliche Bezugsquelle für Geokoordinaten für die gesamte Erde geben müsste, auf der die Werte dann verzeichnet werden können. Die Verwendung von gleichseitigen Sechsecken löst die Probleme der ungleich großen Flächen von Choroplethenkarten und das Problem der Bezugsquelle für Geokoordinaten. Sechsecke sind außerdem Kreisen ähnlicher als Dreiecke, Vierecke oder Einstein-Kacheln, was günstig für die Darstellung von mehreren Datenpunkten innerhalb eines eingegrenzten Gebiets ist.

Der View `hexmap()` beschreibt die Schritte zur Erzeugung der Hexagonalen Dichtekarte. Die Daten werden der Influxdb auf Grundlage der angefragten Ressource (hier: Temperatur) entnommen und in ein df überführt. Der df wird überprüft, ob er Werte für die angefragte Ressource enthält. In Fällen, wo eine Ressource falsch geschrieben wurde, eine nicht existierende Farbskala angefragt wird oder für den ausgewählten Zeitraum keine Messwerte vorliegen, wird eine Hilfsseite gerendert und angezeigt, die den Lernenden eine Auswahl an gültigen Werten vorschlägt. In der Regel finden sich Messwerte in dem df. In dem Fall werden die Messwerte mit Geokoordinaten aus der PostgresDB Tabelle `SenseBoxTable` verknüpft. Danach folgen einige Schritte, in denen der df von nicht benötigten Spalten bereinigt wird, was den Speicherverbrauch reduziert. Danach werden die Zeitpunkte der gemessenen Werte auf volle Stunden gerundet, sowie der Median dieser Werte für jede Stunde berechnet. Diese Schritte dienen einer Zusammenfassung der Werte, was wieder den Speicherverbrauch senkt und die Lesbarkeit in der späteren Darstellung verbessert. Erst danach werden alle Werte auf Stundenbasis aus der Tabelle entfernt, in denen keine Daten übermittelt wurden, da die spätere Darstellung keine nicht numerischen Werten (NaN) erlaubt.

Anschließend werden die Ausreißer in den Daten in Form der oberen und unteren 1 % gefunden und aus dem Datensatz herausgefiltert. Es kommt vor, dass Sensoren unplausible Werte senden. Das Problem wird hier zumindest in seiner extremsten Form abgemildert. Es werden nicht alle fehlerhaften Sensorwerte gefiltert und ist auch nicht Ziel dieser Funktion. Es erleichtert aber die Lesbarkeit in der späteren Darstellung.

Anschließend werden Berechnungen für die Sechsecke durchgeführt. Damit das Diagramm den richtigen Kartenausschnitt zeigt, wird zunächst das Zentrum bestimmt. Danach wird aus dem Radius um das Zentrum (beide Werte kommen aus der Tabelle `SenseBoxLocation`) und dem Erdradius der östlichste und westlichste Längengrad bestimmt. Diese Berechnung korrigiert den Effekt, dass Längengrade in Abhängigkeit von den Breitengraden unterschiedlich weit von einander entfernt sind. Die beiden Längengrade werden anschließend zum df hinzugefügt. Damit wird die Darstellung nach Osten und Westen hin begrenzt. Das bewirkt, dass unabhängig von der Anzahl der später dargestellten Messwerte (und Sechsecke) immer ein gleich großer Kartenausschnitt, mit immer gleich großen Hexagonen angezeigt wird, was die Lesbarkeit und Vergleichbarkeit von unterschiedlichen Darstellungen und Kartenausschnitten verbessert.

Anschließend wird das Diagramm gerendert. Der Kartenausschnitt wird dabei auf das Zentrum ausgerichtet, eine Farbskala wird aus den verfügbaren Farbschemata angewandt („Continuous Color Scales in Python Plotly“, 2025), eine Größe für die Sechsecke wird festgelegt (die Breite ergibt sich aus dem Radius geteilt durch die gewünschte Anzahl). Die Sechsecke werden mit einer Deckkraft von 70 % gezeichnet. Das ermöglicht, dass die Karte darunter weiterhin sichtbar bleibt. Zudem werden die Koordinaten der Messpunkte dunkelrosa eingefärbt. Die Farbe steht zu den meisten Farbskalen in Kontrast, was sicher stellt, dass sie gut sichtbar bleiben. Die Werte der Messpunkte werden hier anschließend passend zu ihrer Zugehörigkeit zu einem der Sechsecke mit dem Median zusammengefasst. Der sich daraus ergebene Wert wird benutzt, um das Sechseck passend einzufärben. Im letzten Schritt werden alle Sechsecke entfernt, die nicht mindestens einen Datenpunkt enthalten. Das Rendering wird nun für jede Stunde in dem ausgewählten Zeitintervall (hier: 48 Stunden) wiederholt. Das Ergebnis davon ist einer Animation der Messwerte über die Zeit.

3.4.3 Berliner Wärmeinseleffekt

Die Karte für den Berliner Wärmeinseleffekt setzt auf der Idee der Berliner Erfrischungskarte (Technologie Stiftung Berlin, 2023) auf. Das Projekt hat eine Plattform entwickelt, die kühle und erfrischende Orte in Berlin aufzeigt. Die zugrunde liegenden Daten entstammen dem Umweltatlas aus dem Geoportal Berlin der Senatsverwaltung für Stadtentwicklung, Bauen und Wohnen (2015).

Der Umweltatlas listet Karten unter anderem zu den Phänomenen bodennahes Windfeld,

Kaltluftvolumenstrom sowie Lufttemperatur auf, die in der Erfrischungskarte kombiniert in einer Karte dargestellt werden. Die Daten für den Kaltluftvolumenstrom liegen für die Zeiten 4:00 und 22:00 Uhr vor. Die Daten für Lufttemperatur liegen für die Zeiten 4:00, 14:00 und 22:00 Uhr vor und stellen die Werte für typische Sommertage dar, die aufgrund fehlender Orographie vor allem die Folge von bebauten und begrünten Flächen in Berlin sind („Lufttemperatur Umweltatlas Berlin“, 2015). Die Karten des Umweltatlas bilden die dargestellten Werte laut Legende für die Uhrzeiten mit konkreten Einheiten ab. Der Kaltluftvolumenstrom im Flächenmittel wird im Bereich > 50 bis $\geq 250 \text{ m}^3/\text{s}$ dargestellt. Die Temperaturwerte werden im Bereich ≤ 15 bis $> 32^\circ\text{C}$ dargestellt („FIS-Broker: Lufttemperatur Daten“, 2025; „FIS-Broker: Winddaten“, 2025). Um Kaltluftvolumenstrom (kühler Wind) und Temperatur auf Stundenbasis verteilt über den Tag anzuzeigen, wurden die fehlenden Werte linear interpoliert. Um Temperatur und Wind vereinheitlicht auf einer Karte darzustellen wurden die interpolierten Werte in 5 Quantile geteilt. In Folge dessen wurden die Einheiten für beide Wertereihen entfernt. Ein Quantil entspricht 20 % des Wertebereichs, was bei Wind für ein Quantil etwa $60 \text{ m}^3/\text{s}$ entspricht und bei Temperatur etwa $4,5^\circ\text{C}$, je nachdem wo die konkreten oberen und unteren Grenzen für Wind und Temperatur angesetzt werden. Beide Werte können nun gemischt dargestellt werden, wie in der Abbildung 8 in der Legende als 5x5 Matrix zu sehen ist. Die in dieser Form aufbereiteten Daten („Rohdaten Erfrischungskarte“, 2021) werden von der Technologie Stiftung Berlin zur Verfügung gestellt.

Zur Erstellung eines stundengenauen Flächenkartogramms (Choroplethenkarte) in den Zeiten von 9:00 bis 21:00 Uhr wird der View `erfrischungskarte(this_time)` verwendet. Neben dem Parameter `request` für den Kontext, wird `this_time` für die Darstellung eines Stundenwerts abgefragt. Bei Funktionsaufruf werden zunächst kontextabhängig die Maße für das Kartogramm gesetzt, Farbwerte für die Einfärbung aller Werte festgelegt und weitere Variablen, die sicher stellen, dass das Diagramm gut auf verschiedenen Bildschirmgrößen ablesbar ist. Die Funktion `load_geojson` lädt bei dem ersten Funktionsaufruf die aufbereiteten Daten aus der zur Verfügung gestellten Quelle und speichert sie dauerhaft zur späteren schnellen Verfügbarkeit lokal im Verzeichnis `refreshing_data/`. Die Funktion `load_data()` ruft diese Funktion auf, worauf hin sie die eben abgerufenen Daten erhält und einen bivariaten Datensatz erstellt. Aus diesem Datensatz wird mit der Funktion `create_bivariate_map()` eine Landkarte erzeugt in der beide Werte gemischt in einem Flächenkartogramm dargestellt werden (Abbildung 8). Die Vorgehensweise über die bivariate Karte stellt sicher, dass alle Mischfarben korrekt dargestellt werden. Bei Verwendung von je einer Ebene pro Datenreihe auf einer Landkarte entscheidet der Webbrowswer über die Darstellung von Mischfarben und Transparenzen, was zu inkonsistent dargestellten Farbwerten führen würde. Zudem erlaubt die Darstellung der Daten in Form einer bivariaten Karte die Reduzierung der Dateigröße ($2 \times 29,6 \text{ MB}$ für die Rohdaten + Metadaten für die Darstellung auf der Karte = $332,1 \text{ MB}$ für das Flä-

chendiagramm) auf 22,73 MB, was relevant ist, wenn die Karte in Volumentarifen und über langsame Verbindungen abgerufen wird. Zusätzlich wird je nachdem, was der anzuzeigende Webbrowser für Standards unterstützt, die übertragene Datenmenge für jede Kartendarstellung durch Kompression mittels gzip oder zstd auf etwa 5,61 bis 7,24 MB reduziert.

3.4.4 Live vor Ort – Sensordaten am Standort

Das Ziel dieses Moduls ist es, die Messwerte aller SenseBoxen an einem Standort anzuzeigen und damit vergleichbar zu machen. Die Funktion wird über den View `show_by_tag()` realisiert. Der View nimmt die Parameter `request`, `box` und `cache_time` entgegen. Über die Query-Parameter in der URL werden die Werte `template_to_use`, `permanent_name` und `tag` übertragen. Der Parameter `region` steuert, welche Region angezeigt werden soll. Es können beliebig viele Regionen in der Tabelle `SenseBoxLocation` angelegt werden. Wenn kein Wert für die Region angegeben wird, wird Berlin als Standard gesetzt. Der Parameter `box` steuert, welche SenseBox angezeigt werden soll. Eine gültige Eingabe ist ein beliebiger Name für eine SenseBox aus der Tabelle `SenseBoxTable`. Wird kein Wert angegeben, wird der Wert auf `all` gesetzt, was stellvertretend für alle, oder beliebige SenseBoxen steht. Der Parameter `cache_time` hat als Vorgabe den Wert 60, was später als 60 Sekunden interpretiert wird. Die Query-Parameter werden in der URL codiert übergeben und steuern unabhängig vom URL-Pfad das Verhalten der Funktion. Mit dem Parameter `tag` wird ein Eintrag aus der Spalte der gültigen GroupTags aus der Tabelle `SenseBoxTable` ausgewählt. Für den Fall, dass der Query-Parameter nicht übergeben wurde, wird der Wert für den GroupTag auf „Humboldt Explorers“ gesetzt. Das ist der Wert, der jeder SenseBox aus dem gleichnamigen Projekt gegeben wird. Zusätzlich dazu tragen die SenseBoxen GroupTags, die einen Hinweis auf den Schulnamen, das Projekt oder allgemein den Standort geben. Über den GroupTag sollen Lernende in der Lage sein, ihre SenseBox schnell aus der Vielzahl der vorhandenen SenseBoxen zu ermitteln und sich die Messwerte davon anzeigen zu lassen.

Der Query-Parameter `permanent_name` ist eine zufällig vergebene und eindeutige Zeichenkette. Der Parameter dient im späteren Verlauf dazu, die einzelnen Zustände der Diagrammdarstellungen voneinander unterscheidbar zu machen. Sollte er nicht vergeben sein, wird ihm ein Wert im Verlauf der Berechnungen zugewiesen. Der letzte entscheidende Parameter `template_to_use` entscheidet darüber, welches Template für die Darstellung auf der Webseite verwendet werden soll. Dieser View ist in der Lage kontextabhängig zwischen dieser Ansicht und der Ansicht *Live im Vergleich – Sensordaten über Standorte hinweg* (Abschnitt 3.4.5) umzuschalten.

Ein weiteres Ziel dieses Moduls ist es die aktuell gemessenen Werte der SenseBoxen anzuzeigen. Wie in „Real time · Issue #40“ (2016) ausgeführt, ist die API nicht in der Lage Daten in Echtzeit zu übermitteln. Bei diesem Konzept würde das DataLab eine Anfrage

an die API senden, welche Sensoren abgefragt werden sollen. Die API würde darauf hin eine dauerhafte Verbindung zum DataLab herstellen (WebSocket) und von sich aus bei Änderungen das DataLab benachrichtigen, so lange diese Verbindung bestehen würde. Statt dessen muss die API regelmäßig nach neuen Daten gefragt werden. Da sie auch dann ihre Datenbank befragen muss, auch wenn keine Änderungen stattgefunden haben, ist diese Art der Bereitstellung für die API und das DataLab ressourcenintensiver.

Um die Messwerte der Sensoren zu erhalten, müssen zwei Abfragen erfolgen. Die erste Anfrage wird mit einer Auflistung von SenseBoxen und den daran angeschlossen Sensoren beantwortet. Danach erfolgt eine zweite Anfrage auf Grundlage der ersten Antwort, um die Sensorwerte zu erhalten.

Für die Abfrage der verfügbaren SenseBoxen wird die Funktion `get_latest_boxes_with_distance_as_df(region, cache_time)` genutzt. Sie ruft die openSenseMap-API ab und speichert das Ergebnis als DataFrame (df) für 60 Sekunden im Cache. Das Speichern für kurze Zeit verhindert, dass die API in sehr kurzen Intervallen abgerufen wird, was die Ressourcen der API schont. In Abbildung 4 ist so eine Antwort von der API zu sehen. Die zweite Anfrage, mit der die Messwerte übertragen werden sollen, wird so angepasst, dass die API nach Möglichkeit nur wenig Werte in ihrer Datenbank ermitteln muss und damit schneller antwortet. Um die Nachfrage zu optimieren, werden zunächst die Einträge in der Spalte `GroupTag` aus der Tabelle `SenseBoxTable` zeilenweise als Einträge einer Liste interpretiert. Das Ergebnis wird im df in der gleichen Spalte gespeichert. Die Mehrzahl der SenseBoxen in dieser Tabelle haben keinen Group-Tag, weshalb diese Zeilen aus dem df entfernt werden. Im Anschluss werden ungültige Einträge, wie leere GroupTags oder GroupTags bestehend aus einem Leerzeichen entfernt.

Als letzten Bereinigungsschritt wird eine Liste aller GroupTags erstellt, wobei Dopplungen ausgeschlossen werden. Eine Funktion prüft, ob der Wert von dem Query-Parameter `tag` (wenn nicht vergeben, dann trägt er den Eintrag „Humboldt Explorers“) in dieser Liste vorkommt. Sollte der Eintrag nicht enthalten sein, dann wird eine Antwort formuliert, die eine Auflistung aller tatsächlich vergebenen GroupTags einhält. Als Ergebnis davon wird eine Webseite gerendert, die eine Auswahl von gültigen Werten präsentiert.

Ist der GroupTag in der Auflistung aller GroupTags enthalten, dann wird mit der Funktion `(get_sensebox_data()` eine zweite Anfrage an die API formuliert, um die aktuellen Messwerte der Sensoren zu erhalten. Da die API für die Beantwortung länger braucht und die Webseite auch für große Gruppen möglichst verzögerungsfrei funktionieren soll, wird auch dieses Ergebnis für 60 Sekunden im Cache zwischengespeichert. Das hat zur Folge, dass pro GroupTag maximal eine Anfrage pro Minute an die API gestellt wird, unabhängig davon, wie häufig die Webseite angezeigt wird. In dem Fall kommen alle Ergebnisse nach der ersten Anfrage aus dem Cache.

Das Ergebnis der Anfrage wird wieder in einem df namens `results` gespeichert. Die

beiden DataFrames `df` und `results` werden zu einer Tabelle vereinigt. Da die Daten in den beiden Tabellen jeweils unterschiedlich strukturiert sind, werden sie beide zeilenweise in Series zerlegt. Aus den Series werden nun die benötigten Werte extrahiert und jeweils in einem Dictionary gespeichert. Dieses enthält den Zeitpunkt der Messung, den Namen des Sensors, den Namen der SenseBox, die GroupTags (inklusive des ausgewählten GroupTags), die Koordinaten des Standorts, den Messwert, die Einheit für die Messung, sowie die Sensor ID. Die Dictionarys werden jeweils in eine Liste geschrieben, aus der wiederum ein neuer DataFrame erzeugt wird. Die Zeichenkette in der Spalte für den Zeitpunkt der Messung wird als ein DateTime-Objekt interpretiert, danach wird die Zeitzone der Messungen von Zulu-Zeit auf die aktuelle Zeitzone korrigiert. Die Zeichenketten aller Messwerte werden als Zahlen interpretiert, danach wird der Mittelwert der Messwerte aller SenseBoxen mit dem gleichen GroupTag zu den jeweiligen Zeitpunkten ermittelt und in die Tabelle geschrieben. Wenn der Parameter `box` den Wert `all` trägt, dann wird durch die Funktion `calculate_centroid()` der Mittelpunkt der Geokoordinaten aller SenseBoxen mit dem gleichen GroupTag ermittelt und in die Tabelle geschrieben. Wenn die Werte der SenseBoxen später auf einer Webseite darstellt werden, dann wird der Mittelpunkt verwendet um auf einer Landkarte anzuzeigen, wo sie in etwa stehen. Gedacht ist diese Funktion, für die Darstellung der SenseBoxen auf einem Schulhof. Es ist dann übersichtlicher nur einmal das Zentrum aller SenseBoxen zu zeigen, statt jede Box einzeln einzzeichnen. Wenn der Parameter nicht `all` ist, dann wird die Liste der SenseBoxen nach dem Namen der SenseBox gefiltert und nur die Zeilen behalten, die den Namen enthalten.

Die anschließende Funktion ermittelt die jeweils aktuellsten Messwerte aus dem DataFrame und speichert diese zusammen mit einem Diagramm, dass im Zuge dieser Funktion gerendert wurde, in der Liste `list_of_dicts_with_rows_and_graphs`. Die letzten Schritte dieses Views sind, die Bezeichnung für den GroupTag in ein URL-kompatibles Format zu bringen und den Wert für `unique_name` auf eine zufällige Buchstabenkombination zu setzen. Durch diesen Trick können bei der Darstellung auf der Webseite ausreichend viele Landkarten durch JavaScript Bibliothek Leaflet gezeichnet werden. Leaflet muss die Karten durch eine Bezeichnung eindeutig von einander unterscheiden können, da es sonst zu Konflikten kommt und Karten weder gelöscht noch aktualisiert werden können. Im letzten Schritt werden alle Listen, berechneten Werte und neu gesetzten Parameter an das Template übergeben, was darauf hin gerendert wird und an den Webbrower zur Darstellung übergeben wird.

Für das Rendering der Webseite wird das Template `base_single_grouptag.html` verwendet. Mit der Django Template Language werden in die statischen html-Zeilen die Werte der Variablen eingesetzt, die durch den View erzeugt wurden. Das Ergebnis davon ist eine statische Webseite. Die dynamischen Elemente, also Bestandteile, die etwas animieren und auf Eingaben reagieren können, werden durch JavaScript, CSS und vor allem durch

htmx (Gross, 2024) erzeugt. So ist es ein htmx-Element, dass nach erstmaligen Laden der Webseite ein Element erzeugt, dass über das Attribut `hx-trigger="every 60s"` in regelmäßigen Abständen eine neue Anfrage an den View auslöst. Die Antwort des Views wird nur in dem Sub-Template `dashboard_single_grouptag.html` gerendert und danach durch htmx in die bestehende Webseite eingebettet. Das Ersetzen eines Teils der Webseite hat den Vorteil, dass der Browser den aktuellen Status der Webseite beibehält und es zu keinen Scroll- oder Flacker-Effekten durch ein Neuladen der gesamten Seite kommt.

Das Template-Fragment von `base_single_grouptag.html` enthält im Bereich `<script> ... </script>` Funktionen, welche die Leaflet-Karte inklusive aller auf ihr angezeigten Elemente und Bedienelemente zeichnet. Das `html`-Element mit der ID `single_location_graph_{{ counter }}` enthält den Bereich, in den die Diagramme gezeichnet werden. Der Skript-Bereich enthält eine Funktion, dass bei einem Neuladen des Sub-Templates `dashboard_single_grouptag.html` das bisherige Diagramm aus dem Speicher entfernt wird und mit dem Neuen ersetzt wird. Diese Funktion verhindert, dass die Diagramme im Speicher verbleiben und die Webseite damit nach einiger Zeit instabil wird.

3.4.5 Live im Vergleich – Sensordaten über Standorte hinweg

Das Modul zur Darstellung beliebiger SenseBoxen zum Vergleich ihrer Messwerte setzt auf den gleichen View auf, wie das Modul *Live vor Ort – Sensordaten am Standort*, das in Abschnitt 3.4.4 beschrieben wird. Auch hier ist anhand von GroupTags und SenseBox-Namen eine Auswahl möglich. Die Unterschiede in der Berechnung und der Darstellung durch den View `show_by_tag()` werden durch den Query-Parameter `template_to_use` gesteuert. Es wird zur Darstellung ein anderes Template verwendet (`compare_dashboard.html` statt `dashboard_single_grouptag.html`) und es wird zusätzlich zu dem ursprünglichen Kontext die Variable `permanent_name` auf eine zufällig generierte Zeichenkette gesetzt, so wie es auch vorher für die Variable `unique_name` gemacht wurde. Die Variable `permanent_name` stellt sicher, dass die Ansichten, die später generiert werden, erhalten bleiben, wenn sie den View nach neuen Werten fragen und bei der Antwort aktualisiert werden. Die Templates funktionieren nach dem gleichen Prinzip wie im Abschnitt 3.4.4, sie führen nur zu einer anderen Darstellung. Die einzige wesentliche Änderung ist, dass es nun möglich ist, beliebig viele Diagramme gleichzeitig von unterschiedlichen Standorten zu zeigen.

3.4.6 Interaktive Datenexploration

Die Erstellung aller Module und deren Diagramme wird über Pfadangaben in der URL oder durch Query-Parameter gesteuert. Dadurch ist es möglich ohne die Programmierung und die Datenverarbeitung im Hintergrund zu verstehen und zu verändern, die Darstel-

lung der Diagramme zu beeinflussen. Über den View `single()` wird eine Ansicht generiert, die es ermöglicht eigene hexagonale Dichtekarten im Stil von Abschnitt 3.4.2 zu zeichnen. Dazu wird die InfluxDB befragt, welche Sensoren in den letzten zwölf Stunden Messwerte übertragen haben. Diese werden als aktiv betrachtet. Alle anderen Sensoren werden nicht weiter berücksichtigt. Danach werden folgende Kontextvariablen definiert: `ressource_path` (= verfügbare Sensoren), `start_time` (= Beginn des Zeitfensters), `colorscale` (= Liste aller verfügbaren Farbskalen), `map_style` (= Stil der zugrundeliegenden Landkarte) und `resolution` (= Größe der Hexagone). Anschließend werden Vorgabe-werte definiert, welche die Variablen bei dem ersten Aufruf der Webseite tragen sollen. Das Ergebnis davon wird über den Session Cookie gespeichert. Jeder Browser bekommt bei dem ersten Aufruf der Webseite eine Session zu gewiesen, die vom Browser in Form einer Session ID (`sessionid`) in Form einer langen Zeichenkombination als Session Cookie abgespeichert wird. Parallel dazu wird im Session-Backend von Django (in dem Fall in der Datenbank Redis) die gleiche Session ID abgelegt. Diese ID kann mit weiteren Informationen verknüpft werden. Üblicherweise ist das der Login-Status. In diesem Fall werden die Kontextvariablen in der Variable `last_params` hinterlegt.

Die Webseite die darauf hin aus dem Template `customizer.html` gerendert wird enthält Drop-Down Elemente, mit den Kontextvariablen, die der View erzeugt hatte. Hier lässt sich auswählen, wie das Diagramm gezeichnet werden soll. Sobald ein Eintrag ausgewählt wurde, fragt eine `htmx`-Funktion den View `url_string_generator()` an. Dieser fragt die Kontextvariablen aus der Variablen `last_params` aus dem Session Cookie ab. Danach wird abgefragt, welche Kontextvariable über das Drop-Down Element ausgewählt wurde und legt das Ergebnis in `new_params` ab. Da beide Werte Python Dictionarys sind, lassen sie sich einfach zu einem Dictionary in der Variable `params` vereinen. Der Wert von `params` wird nun als `last_params` in den Webseiten Cache geschrieben. Danach wird eine URL aus den Kontextvariablen erzeugt. Der View gibt im letzten Schritt einen `html`-Button zurück, der auf der Webseite neben den Drop-Down Elementen gezeichnet wird. Durch Auswahl des Buttons wird der View `hexmap()` aufgerufen. Der View rendert ein Diagramm anhand der ausgewählten Werte und zeigt ihn in einem neuen Tab in dem Browser an.

4 Ergebnisse

4.1 Messstellen-Karte

Die *Messstellen-Karte* (Abbildung 5) ist zugleich die Startseite des DataLabs. Es wird eine Landkarte angezeigt, auf der mit blauen Markern die Standorte von SenseBoxen markiert sind. Auf der linken Seite befindet sich eine Tableiste mit verschiedenen Symbolen. Diese Symbole werden im Folgenden von oben nach unten gelesen erläutert.

An erster Stelle findet sich ein Symbol (drei horizontale Linien), welches auf eine Übersichtsseite mit beliebig ausfüllbaren Informationen verweist. Hier wäre die Einrichtung einer Startseite möglich, auf der für Lernende oder Lehrkräfte ein Begrüßungstext und Bedieninformationen angezeigt werden können. Das zweite Symbol (Thermometer) verweist auf eine *Hexagonale Dichtekarte*, auf welcher die Temperatur der letzten 48 Stunden in der Region dargestellt wird. Das dritte Symbol (vernebelte Sonne) verweist auf eine *Hexagonale Dichtekarte*, auf welcher die Feinstaubwerte der letzten 48 Stunden in der Region dargestellt werden. Das vierte Symbol (Erfrischungsgetränk) führt zur Darstellung des *Berliner Wärmeinseleffekts*. Das fünfte Symbol eines tags (bzw. eines Labels/ eines Etiketts) führt zur *Live Ansicht vor Ort*. Das sechste Symbol (zwei aufeinander verweisende Pfeile) führt zur *Live Ansicht für einen Vergleich zwischen mehreren Standorten*. Ganz unten in der Tableiste findet sich das Symbol eines Zahnrads, hinter diesem befindet sich der Tab, der zu der „Admin-Ansicht“ (dem Django Panel) und dem „CMS“ (Wagtail) führt. Zusätzlich werden hier Links angezeigt, die zu der *Interaktiven Datenexploration* sowie der Einzel darstellung aller aufgeführten Module hinter einer eigenen URL (etwa zur Einbettung auf anderen Webseiten mittels eines iFrames) führen.

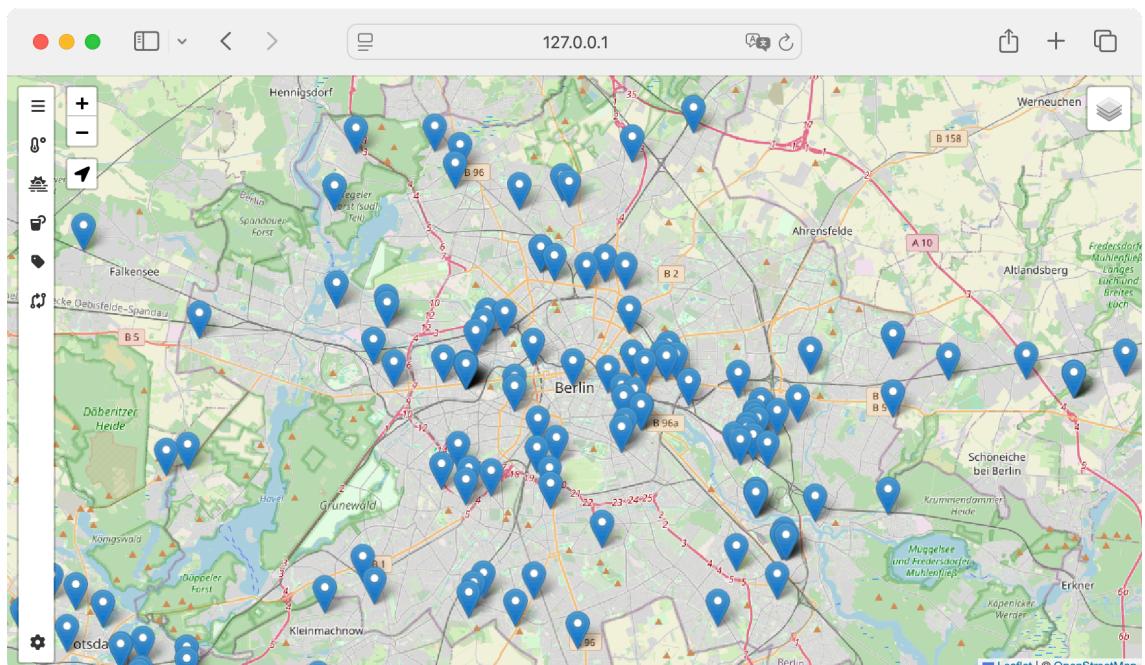


Abbildung 5: Messstellen-Karte mit blauen Markierungen für Messstellen.

Jeder Tab, der sich durch Anklicken eines der Symbole öffnet, zeigt oben in einer blauen Leiste den Titel des Moduls an, der möglichst selbsterklärend für die Funktion steht. Rechts davon befindet sich ein gängiges Schließen-Symbol („x“).

Auf der Startseite/ *Messstellen-Karte* befinden sich im linken oberen Bereich der Seite (rechts neben der Tableiste mit den Symbolen) drei weitere Symbole, um die Karte zu skalieren und um den eigenen Standort zu bestimmen. Die Skalierung kann auch mit einer Zwei-Finger-Geste auf Touchscreens oder mit dem Mausrad erfolgen. Wenn die eigene

Position bestimmt wird, erfolgt eine fließende Vergrößerung des eigenen Standorts auf der Karte.

Oben rechts auf der Karte befindet sich ein Symbol, über das sich die Darstellung der Karte ändern lässt. Es lässt sich von der Straßenansicht auf Satellitenfotos wechseln. In beiden Darstellungen lassen sich 3D-Darstellungen von Gebäuden ein- und ausblenden. Unten rechts auf der Karte werden passend zur Auswahl die Quellen des Kartenmaterials zitiert.

Die blauen Marker werden vom Zentrum Berlins (ungefähr bei 52.516221 °N, 13.3992 °E) ausgehend in einem Radius von 30 Kilometern angezeigt. Diese Werte können über das „Admin Panel“ (hinter dem Zahnrad-Symbol) über den Eintrag SenseBox Location beliebig angepasst werden.

Wenn ein blauer Marker auf der Messstellen-Karte ausgewählt wird, öffnet sich ein schwebendes Fenster in Form einer kleinen Sprechblase. In dieser wird der Name der Messstation und ein Link zum Anzeigen der Daten („Zeige Daten“) dargeboten, außerdem führt ein „Link zur Box“ direkt zur Anzeige der SenseBox auf der openSenseMap, der Webseite des Herstellers.

Wenn die Option „Zeige Daten“ ausgewählt wird, dann öffnet sich ein Modal (eine Art vor der Karte schwebendes Fenster) in dem vertikal übereinander angeordnet die Diagramme für jeden Sensor dieser SenseBox angezeigt werden. Die Darstellung orientiert sich an den Erkenntnissen, die in den Arbeiten von Lachmayer et al. (2007) und Schnotz (1992, 1997) zusammengefasst wurden. Über den Diagrammen wird als Überschrift die Bezeichnung der SenseBox angezeigt. Jedes Diagramm hat eine andere Färbung, um die Diagramme besser voneinander unterscheidbar zu machen. Die Achsen sind mit der Bezeichnung der Variable und der passenden Maßeinheit in Klammern dahinter versehen. Die Höhe der Diagramme ist so gewählt, dass die Beschriftung auf der Y-Achse immer vollständig angezeigt wird und lesbar ist. Alle Diagramme haben untereinander die gleiche zeitliche Einteilung. Die Skalierung der Y-Achse passt sich automatisch an, damit alle Werte sinnvoll dargestellt werden können. Es werden dünne horizontale und vertikale Linien eingezeichnet, um das händische Ablesen von Werten zu vereinfachen. Die Darstellung der Werte erfolgt in einem Liniendiagramm, sofern die Werte mit einem maximalen Abstand von 5 Minuten erhoben wurden. Ist die Zeitspanne zwischen den Messungen länger als 5 Minuten, so werden die Werte als Streudiagramm darstellt. Diese Unterscheidung findet statt, damit bei unzusammenhängenden Daten nicht der Eindruck erweckt wird, dass kontinuierlich Werte gemessen wurden (obwohl hier tatsächlich nur wenige Datenpunkte vorliegen, die lediglich mit einer Linie verbunden wurden). Je nach zeitlichem Abstand zwischen den Messdaten können beide Diagrammtypen auch in gemischter Form auftreten.

Die Diagramme sind auch interaktiv nutzbar: Es lassen sich gezielt Intervalle auf der X-Achse auswählen, wodurch nur diese angezeigt werden. Es lassen sich auch rechteckige Bereiche auswählen, wodurch diese dann vergrößert dargestellt werden. Neben dem hän-

dischen Ablesen von Datenpunkten über die Hilfslinien, lassen sich auch Punkte auf den Graphen auswählen, wobei dann daneben eine farbige Sprechblase mit dem Y-Wert und der Bezeichnung des Sensors angezeigt wird. Zusätzlich wird auf der X-Achse die dazu passende Uhrzeit ebenfalls in einer Sprechblase angezeigt. Oben rechts neben der Überschrift gibt es Symbole, um die Ansicht zu vergrößern oder zu verkleinern, sowie die Ansicht auf den Ausgangswert zurückzustellen (Haus Symbol).

Die Diagramme für Feinstaubmessungen (PM_{10} und $PM_{2.5}$) zeigen zusätzlich gesetzliche Grenzwerte in Form einer horizontalen gestrichelten Linie an (Abbildung 6). Wenn der Grenzwert überschritten wird, dann wird der entsprechende Bereich des Graphen rot hinterlegt. Die Grenzwerte basieren auf der Richtlinie 2008/50/EG des Europäischen Parlaments und des Rates vom 21. Mai 2008 über Luftqualität und saubere Luft für Europa (2008), die zum Zeitpunkt der Erstellung dieser Arbeit einen Jahresmittelwert von 40 g/m^3 für PM_{10} und 25 g/m^3 für $PM_{2.5}$ festlegt. Laut Angaben des Umweltbundesamtes (Wilke, 2015) soll der Grenzwert für $PM_{2.5}$ ab dem Jahr 2030 auf 10 g/m^3 abgesenkt werden.

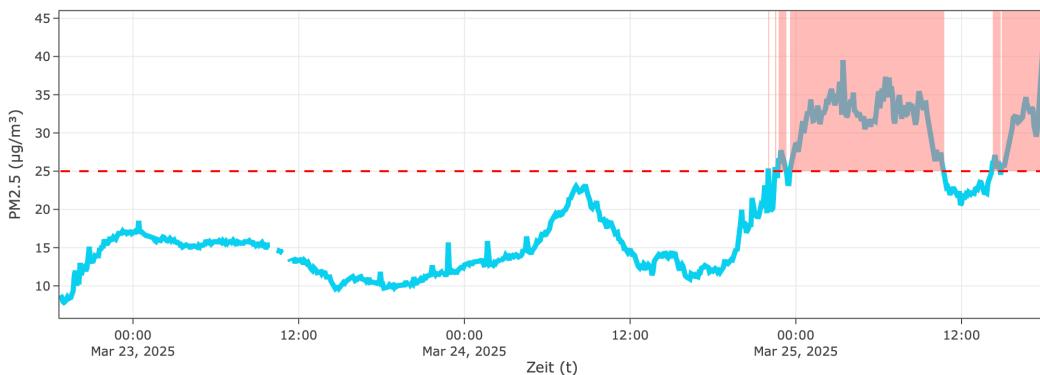


Abbildung 6: Liniendiagramm, wie es im DataLab dargestellt wird.

Das Modal mit der Anzeige aller Sensorwerte der SenseBox lässt sich schließen, indem oben rechts das Schließen-Symbol, unten rechts der Schließen-Knopf oder ein Bereich außerhalb des Modals ausgewählt wird.

4.2 Hexagonale Dichtekarte

Die beiden *hexagonalen Dichtekarten* zu Temperatur und Feinstaubwerten (Abbildung 7) lassen sich auf der Startseite/ *Messstellen-Karte* in der Tableiste auf der linken Seite über das Symbol eines Thermometers oder das Symbol einer vernebelten Sonne (Sonne mit horizontalen Linien im unteren Bereich) öffnen. Das Thermometer Symbol führt zur „Temperatur im Verlauf der letzten 48 Stunden“, das Symbol der vernebelten Sonne zur Ansicht der „Feinstaubwerte im Verlauf der letzten 48 Stunden ($PM10$)“.

Bei den Dichtekarten werden jeweils die Messwerte aus einem sechseckigen Bereich auf der Karte aggregiert und als Farbwert für einen ausgewählten Zeitpunkt dargestellt, sofern

der Bereich mindestens einen Sensor enthält, der Daten gesendet hat. Der Farbwert des Bereichs kann über die Legende auf der rechten Seite einem Sensorwert zugeordnet werden. Befinden sich mehrere Sensoren innerhalb desselben hexagonalen Bereichs, so wird für jeden Zeitpunkt der Median der erfassten Messwerte dargestellt. Die Wahl des Medians als statistisches Maß sorgt dafür, dass einzelne Ausreißer den dargestellten Wert weniger stark beeinflussen als beim arithmetischen Mittel.

Unter der Karte befindet sich ein Bedienfeld mit einer Start und Stopp Taste. Rechts daneben wird mit der Bezeichnung „frame“ der dargestellte Zeitpunkt benannt. Darunter befindet sich ein Zeitstrahl mit Beschriftung. Jede Stunde darin wird mit einem Strich markiert. Wenn das Diagramm geöffnet wird oder durch Auswahl des Start Knopfes wird eine Animation abgespielt. Dabei werden die sechseckigen Bereiche für jede Stunde neu entsprechend der Messwerte eingefärbt. Der Zeitstrahl enthält auch einen Regler, der manuell verschoben werden kann, wenn ein bestimmter Zeitabschnitt angezeigt werden soll.

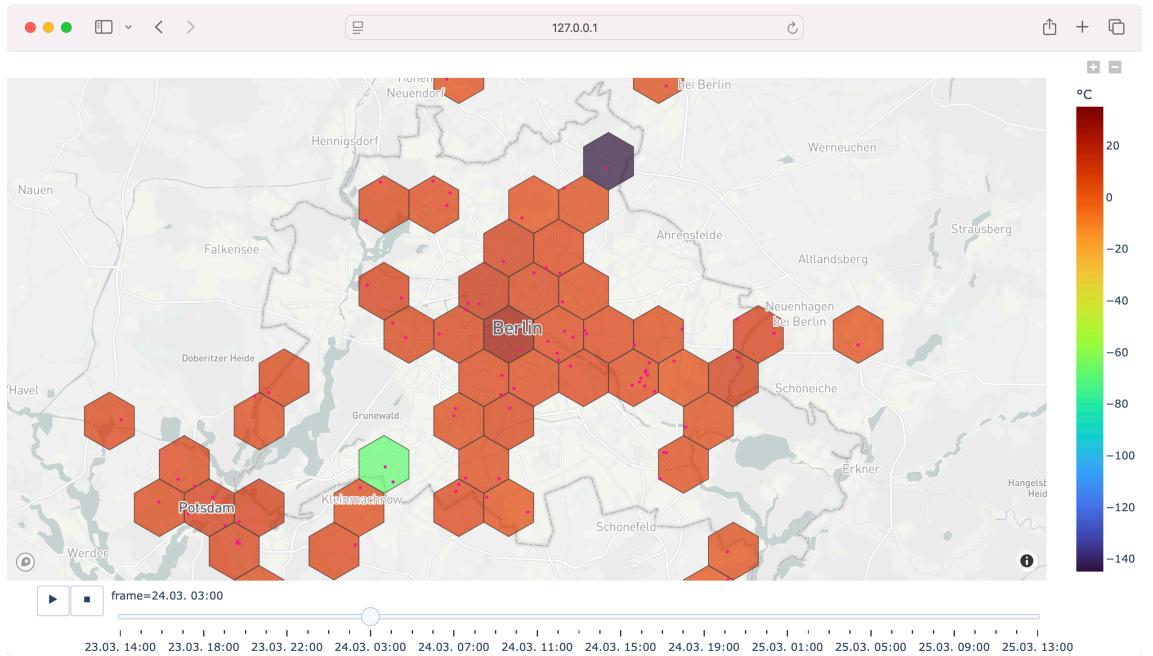


Abbildung 7: Darstellung von Temperaturwerten in Form von Sechsecken auf einer Landkarte in einer Animation über die Zeit.

Indem die Maus auf eines der Sechsecke bewegt wird, oder durch Auswahl mittels Tippen auf einem Touchscreen, lassen sich die Rohdaten des Bereichs anzeigen. Die Karte lässt sich auf verschiedene Arten skalieren. Bei der Bedienung mit der Maus kann oben rechts in der Karte die Vergrößerung verändert werden. Auf einem Touchscreen kann zusätzlich mit einer Zwei-Finger-Geste die Karte angepasst werden. Durch Druck mit der sekundären Maustaste (die rechte Taste bei einer Rechtshänder Maus) oder mit zwei Fingern auf einem Touchscreen, lässt sich die Karte auch drehen und neigen.

Die Karte im Hintergrund ist bewusst dezent und wenig farbenfroh gewählt, damit die farbigen Sechsecke einen starken Kontrast zur Karte bilden. Die Messorte sind mit einem

hellen violett Ton eingefärbt. Durch die Farbwahl entsteht zu den meisten Farben, die die Sechsecke bekommen können, ein guter Kontrast. Die Sechsecke werden dazu mit einer Deckkraft von 70 % gezeichnet, damit die Messorte und die Karte weiterhin sichtbar bleiben. Unten rechts auf der Karte hinter dem i-Symbol werden die Quellen des Kartenmaterials in Abhängigkeit zur Darstellung eingeblendet.

Manche Sensoren liefern unplausible und fehlerhafte Werte (manche Sensoren zeigen etwa die Temperatur von mehreren Millionen Grad Celsius an). Um die Legende und die Farbgebung der Sechsecke durch solche extremen Werte nicht durch zu starke Verzerrung wertlos werden zu lassen, werden die oberen und unteren 1 % der Messergebnisse automatisch verworfen. Durch diese Art der Datenbehandlung werden die meisten fehlerhaften Sensorwerte trotzdem angezeigt, um den Lernenden die Möglichkeit zu geben über die Gründe für diese Werte nachzudenken (Messfehler, ungünstige Platzierung des Sensors, Defekte, ...).

4.3 Berliner Wärmeinseleffekt

Hinter dem Symbol eines Erfrischungsgetränks (Trinkglas mit Zitrone) verbirgt sich die Darstellung der Karte zum *Berliner Wärmeinseleffekt*. Die Karte basiert weitestgehend auf der Idee der Berliner Erfrischungskarte (Technologie Stiftung Berlin, 2023), daher wurde das Symbol gewählt. Die Erfrischungskarte basiert wiederum auf Daten des Geoportals Berlin (Senatsverwaltung für Stadtentwicklung, Bauen und Wohnen, 2015). Die Daten der Erfrischungskarte werden in diesem Modul zur Bewertung des Wärmeinseleffekts neu interpretiert und dargestellt.

Die Karte (Abbildung 8) zeigt sehr kleinteilig in Form eines Flächenkartogramms die Wind- und Temperaturwerte eines durchschnittlichen Berliner Sommertags an. Beide Werte können auch gemeinsam in einem Gebiet dargestellt werden. In dem Fall werden Mischfarben für die Darstellung gewählt. Die Legende in der oberen rechten Ecke erklärt die Zusammenhänge beider Werte.

Die Auswahl der Farben wurde so gewählt, dass die Effekte möglichst gut sichtbar werden und sich intuitiv erschließen lassen. Zudem sollten die Mischfarben sinnvoll interpretierbar sein.

Die Karte lässt sich durch die Symbole oben rechts über der Karte, mit dem Mausrad oder durch eine Zwei-Finger-Geste auf Touchscreens skalieren. Das Haus-Symbol oben rechts setzt die Ansicht zurück. Die Karte kann mit der rechten Maustaste oder mit den Fingern auch geneigt und gedreht werden. Einzelne farbige Gebiete können auf der Karte ausgewählt werden. Es werden dann die Werte in Form einer Sprechblase angezeigt, die zur Einfärbung geführt haben. Die Gebiete sind durchnummeriert und jedes Gebiet hat einen Matrix-Value (MV). Diese beiden Werte hatten einen Zweck bei der Entwicklung

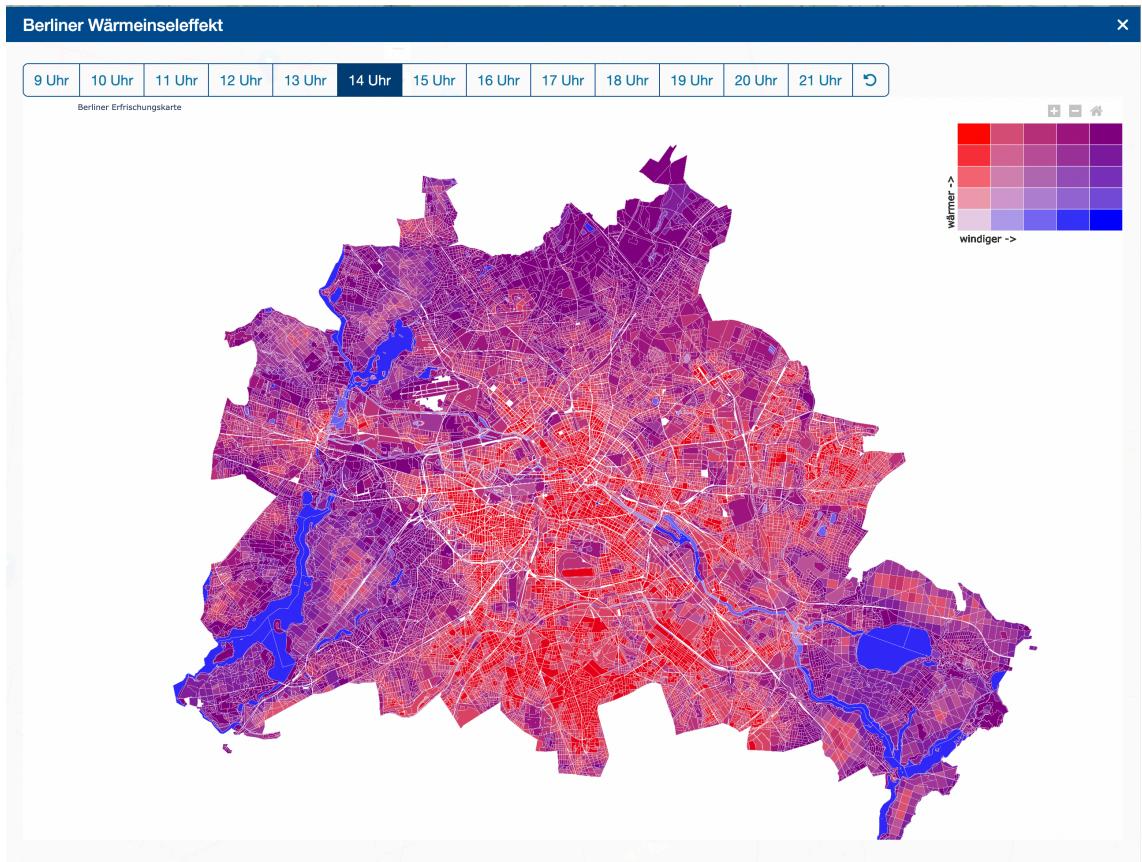


Abbildung 8: Berliner Wärmeinseleffekt um 14 Uhr an einem typischen Sommertag.

der Karte, sie können später auch entfernt werden, wenn das Projekt weiter entwickelt wird.

Über der Karte befinden sich Auswahllemente, mit der die Darstellung für einzelne Uhrzeiten zwischen 9:00 und 21:00 Uhr gewählt werden kann. Ganz rechts in der Leiste kann mit dem rotierenden Pfeil ein GIF abgespielt werden, dass die Darstellungen aller Uhrzeiten des Tages in einer wiederholenden Animation darstellt. Oben links in dem GIF wird die Tageszeit in Form einer sich drehenden Scheibe dargestellt (Abbildung 9).

Das GIF wurde soweit wie möglich komprimiert, ohne dass die Darstellung darunter leidet (Kompressionsartefakte, Verzerrungen, Beschriftung wird unleserlich, ...). Zusätzlich erfolgt die Einbindung der Grafik auf eine Weise, die dem Webbrowser erlaubt die Grafik zu cachen, damit sie nicht bei jedem Seitenaufruf neu übertragen werden muss.

4.4 Live vor Ort – Sensordaten am Standort

Über das Symbol eines tags (eines Labels/ Etiketts) öffnet sich das Modul, mit dem *Live-Sensordaten von bestimmten Standorten* angezeigt werden können. Hierfür ist relevant, dass jede SenseBox nicht nur einen eigenen spezifischen Namen hat, sondern auch noch mit einem oder mehreren GroupTags versehen werden kann. GroupTags können z.B. darüber Auskunft geben, zu welchem Projekt oder zu welcher Gruppe eine SenseBox gehört.

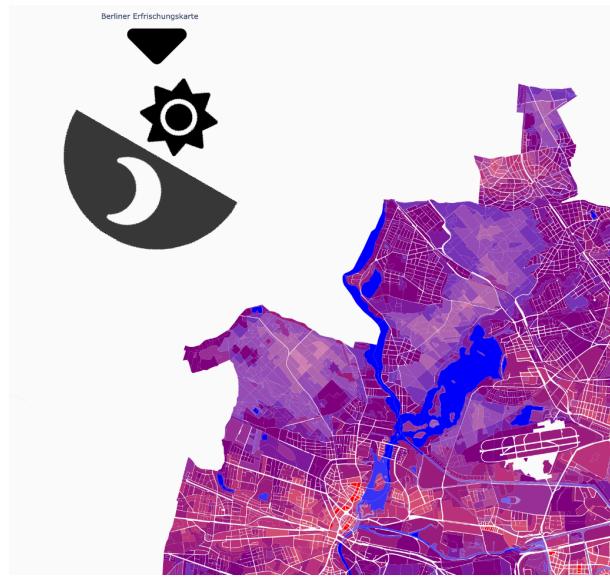


Abbildung 9: Erfrischungskarte mit Animation (Ausschnitt).

So tragen beispielsweise alle SenseBoxen, die im Rahmen der *Humboldt Explorers* zum Einsatz kommen, den GroupTag „Humboldt Explorers“.

Wird das Modul mit den Live-Sensordaten geöffnet, so wird zunächst automatisiert danach gesucht, ob es tagesaktuelle Live-Daten von einer oder mehreren SenseBoxen mit dem GroupTag „Humboldt Explorers“ gibt. Sofern dies der Fall ist werden diese Sense-Boxen sofort angezeigt.

Auf der rechten Seite kann mit dem </>-Symbol diese Ansicht auf einer anderen Webseite in einem iFrame eingebettet werden.

Für den Fall, dass aktuell keine Daten von Sense-Boxen mit dem GroupTag „Humboldt Explorers“ gesendet werden, erscheint die Meldung „Heute wurden noch keine Werte für Humboldt Explorers gemeldet. Wie wäre es stattdessen mit einem von denen hier?“.

Mittels eines Drop Down Menüs („Gruppe auswählen“) können nun auch Sense-Boxen mit beliebigen anderen GroupTags geöffnet werden (etwa die des „Futurium“). In der Auswahl sind bei weitem nicht alle SenseBoxen dabei, da nur eine relativ geringe Anzahl GroupTags verwendet. Wird nun ein GroupTag aus dem Drop-Down Menü ausgewählt, so verweist dieser üblicherweise auf eine Auswahl an mehreren SenseBoxen. Zunächst werden alle Boxen mit dem gewählten GroupTag angezeigt, wobei der Button mit der Beschriftung „Alle Werte“ dunkel erscheint. Daneben werden alle SenseBoxen mit Namen aufgelistet, die diesen GroupTag tragen. Sie können an dieser Stelle ausgewählt werden.

Bei der Anzeige von allen SenseBoxen (Abbildung 10) wird direkt darunter eine Karte im Stil von der Messstellen-Karte angezeigt. Es sind die gleichen Bedienelemente sichtbar. Doch statt alle Messstellen anzuzeigen, wird nur der geografische Mittelpunkt aller SenseBoxen mit dem ausgewählten GroupTag angezeigt. Diese Darstellung wurde vor dem

Hintergrund gewählt, da diese Anwendung für ein Lernmodul auf einem Schulhof gedacht ist und die Karte auf diese Weise nicht durch viele eng beieinander stehende Messstellen unübersichtlich wird.

Unter der Landkarte werden passend zur Anzahl der Sensoren Kacheln angezeigt. Oben in jeder Kachel gibt es ein beschreibendes Symbol, sowie darunter die Bezeichnung für den Sensor. Die Bezeichnung ist genormt, so dass auch bei verschiedenen Schreibweisen die Bezeichnung immer konsistent gleich bleibt. Grundlage dafür ist die Tabelle **Sensors Info Table** im „Admin Panel“. Über diese Tabelle wird auch sichergestellt, dass die Einheiten in allen Diagrammen stets in der korrekten Schreibweise erscheinen.

Unter dem Namen für den Sensor wird der Median aller Sensoren mit der gleichen Bezeichnung für den aktuellen Zeitpunkt dargestellt. Direkt darunter wird pro Sensor ein Diagramm nach den gleichen Regeln wie bei der Messstellen-Karte gezeichnet. Die Messwerte jeder SenseBox werden in einer eigenen Farbe zur besseren Unterscheidbarkeit als Graph eingezeichnet. Die Diagramme lassen sich mit den gleichen Methoden anpassen, wie sie auch schon in den vorherigen Modulen beschrieben wurden. Wie in der Abbildung zu sehen ist, kann auch hier zum Ablesen der Werte ein Bereich markiert werden, wodurch Messwerte farblich passend direkt auf dem Graphen und der Zeitpunkt auf der X-Achse angezeigt werden.

Im Auswahlmenü über der Landkarte kann eine einzelne SenseBox aus der Gruppe gewählt werden. Daraufhin wechselt die Anzeige zur Darstellung von Diagrammen von nur dieser Box. Die Darstellung der Kacheln enthält dann den Namen der SenseBox.

Der aktuelle Messwert jedes Sensors wird in großer Schrift und darunter der Median aller Boxen in kleiner Schrift dargestellt. Der Median wurde gewählt, weil auf diese Weise einzelne Ausreißer-Werte nicht so stark verzerrend wirken können. Wenn die Box exakte Koordinaten hat, wird der Standort entsprechend auf der Karte markiert.

Die Ansicht für die Werte und die Diagramme werden alle 60 Sekunden mit aktuellen Daten neu gezeichnet. Um die Last für den API-Server der openSenseMap möglichst gering zu halten, wird sichergestellt, dass maximal eine Anfrage pro Minute gestellt wird. Das Ergebnis wird im Cache des DataLabs gespeichert. Dieser enthält dann alle Werte, die aktuell benötigt werden.

4.5 Live im Vergleich – Sensordaten über Standorte hinweg

Das Symbol der beiden aufeinander verweisenden Pfeile öffnet das Modul für die Anzeige von *Sensordaten über Standorte hinweg*. In diesem Modul können, wie der Name es andeutet, beliebig viele SenseBoxen ausgewählt und untereinander verglichen werden.

Die Ansicht zeigt zunächst nur ein einfaches Plus-Symbol. Wenn es ausgewählt wird, holt das DataLab eine Liste aller verfügbaren GroupTags und den dazugehörigen SenseBox-

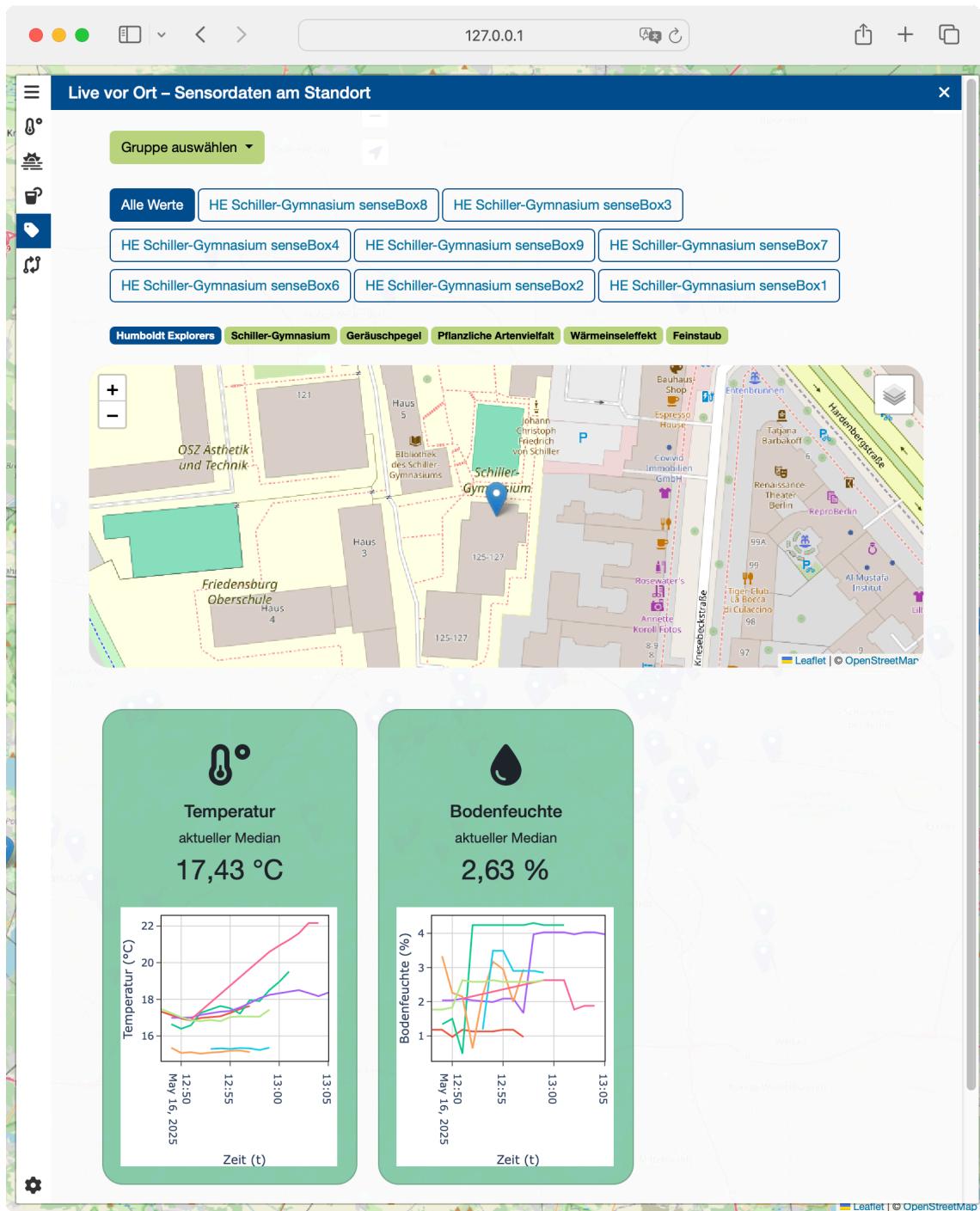


Abbildung 10: Darstellung von *Live vor Ort – Sensordaten am Standort* während einer laufenden Unterrichtseinheit.

Namen vom API-Server. Dieser Prozess kann einige Sekunden dauern. Um anzuzeigen, dass eine Eingabe erfolgt ist, dreht sich das Symbol.

Sobald der API-Server die Daten geschickt hat, wird eine Auswahl an möglichen Optionen angeboten. Wenn innerhalb von einer Minute eine konkrete SenseBox ausgewählt wurde, wird die Kachel mit den Daten der Box gefüllt. Wird länger gewartet, dann wird zunächst wieder eine neue Anfrage gestellt, bevor etwas auf der Kachel angezeigt wird.

Wenn eine konkrete SenseBox ausgewählt wurde, wird auf der Kachel oben der Name der Box als Überschrift gezeigt. Darunter erscheint die Karte mit den bereits vorgestellten Optionen. Im Zentrum ist die Markierung mit dem Standort der Box zu sehen. Darunter folgt eine Auflistung aller Sensoren der Box. Neben jedem Sensornamen wird der aktuelle Wert mit Einheit gezeigt. Es kann ein Eintrag ausgewählt werden, worauf hin sich dieser Bereich vergrößert und ein Diagramm über alle Werte seit Mitternacht des aktuellen Tages gezeigt wird. Auch diese Darstellung entspricht der, wie sie weiter oben schon beschrieben wurde.

Unten auf der Kachel kann über das </>-Symbol eine URL für diese Ansicht erzeugt werden, womit diese Kachel auf einer anderen Webseite eingebettet werden kann. Rechts neben der ersten Kachel wird ein weiteres Plus-Symbol angezeigt, mit dem eine weitere Kachel der Ansicht hinzugefügt werden kann. Oben rechts hat jede Kachel einen Knopf um sie aus der Ansicht zu entfernen.

Alle ausgewählten Kacheln zeigen alle 60 Sekunden die aktuellen Werte an. Durch die Auswahl von Sensoren des gleichen Typs, ist ein Vergleich von mehreren Boxen über Standorte hinweg möglich (Abbildung 11).

4.6 Interaktive Datenexploration

Das Modul für die *interaktive Datenexploration* verbirgt sich hinter dem Zahnrad-Symbol links unten in der Tableiste. Die Darstellung ist weniger für den Schulunterricht und mehr für die Arbeit mit Studierenden gedacht, weshalb diese Position gewählt wurde.

Hinter dem Zahnrad-Symbol sind auch die Zugänge zum Admin Panel und zum CMS zu finden. Unten gibt es ein Drop-Down Menü mit „Direkt URLs zu den Grafiken“. Diese können verwendet werden, um entweder die Grafiken direkt anzuzeigen oder in iFrames einzubetten.

Mit diesem Modul ist es möglich eigene hexagonale Dichtekarten zu gestalten (Button „Eigene Diagramme erstellen“. Dazu werden in einer Auswahl alle verfügbaren Parameter in ausklappbaren Listen präsentiert (Abbildung 12).

Die verfügbaren Parameter sind: „Sensor“ (hier werden alle Sensortypen des aktuellen Tages aufgelistet), „Zeitfenster“ (wie weit die Messwerte vom aktuellen Zeitpunkt begin-

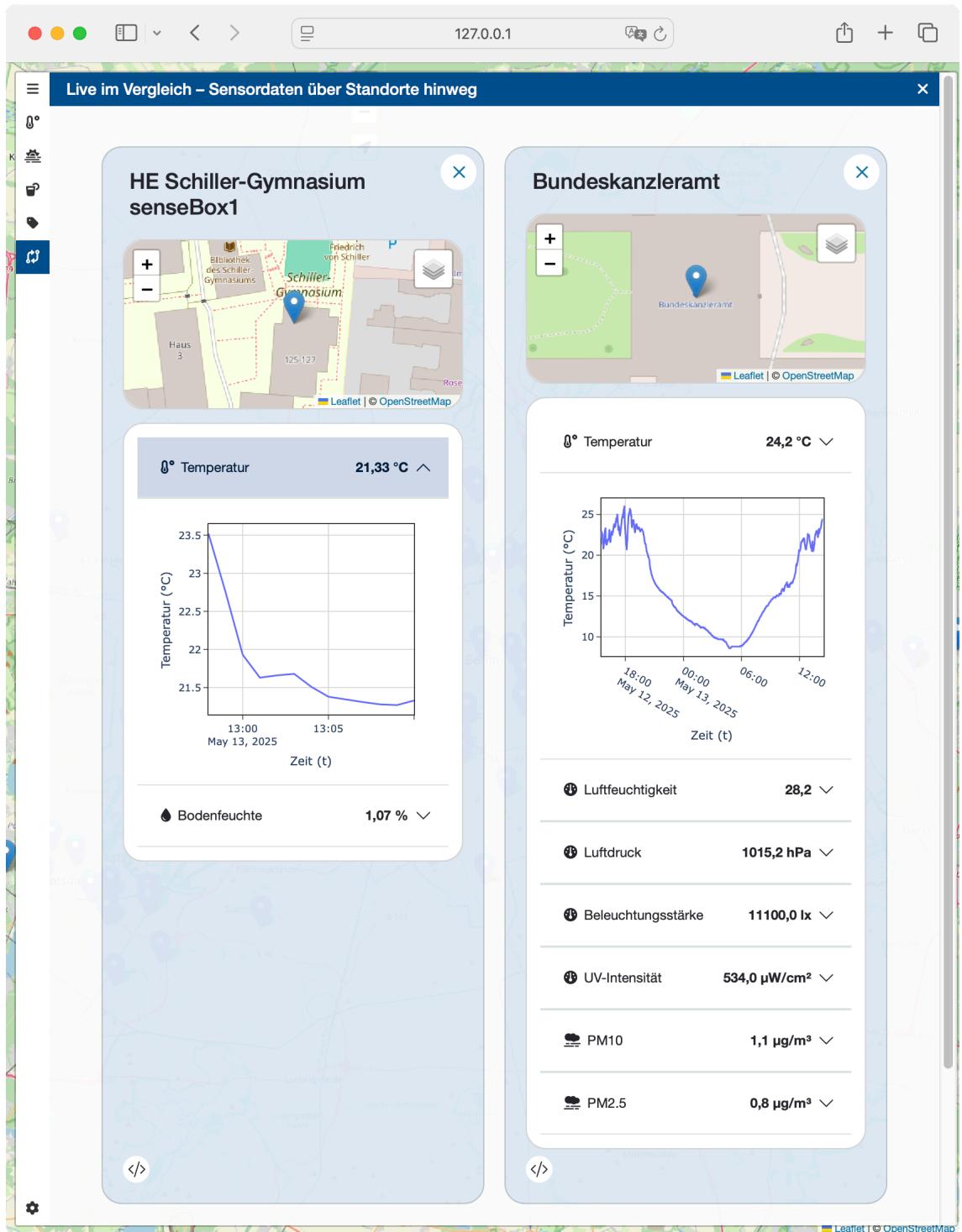


Abbildung 11: Darstellung von zwei Standorten mit dem Modul *Live im Vergleich – Sensordaten über Standorte hinweg*.

nend in die Vergangenheit angezeigt werden sollen), „Farbschema“ (eine Auflistung aller Optionen verbirgt sich hinter dem Link in dem Feld „Info“), „Kartentyp“ (zugrundeliegende Landkarte) und „Auflösung“ (Größe der Hexagone). Einige der Parameter enthalten darunter ein ausklappbares Informationsfeld mit Bedienhinweisen oder weiterführenden Informationen.

Durch Auswahl von Werten aus den ausklappbaren Listen wird im Hintergrund eine URL generiert. Diese URL wird hinter den auf der rechten Seite liegenden Knopf mit dem Dreieckssymbol gelegt. Wenn der Knopf gedrückt wird, zeichnet das DataLab ein Diagramm mit den ausgewählten Parametern. Die URL wird in der Adresszeile des Webbrowsers angezeigt und kann verwendet werden, um diese Darstellung auf einer anderen Webseite mittels iFrame einzubetten.



Abbildung 12: Auswahlmöglichkeiten zur Erstellung eigener Diagramme.

5 Diskussion

Das übergeordnete Ziel der vorliegenden Arbeit bestand darin, eine webbasierte Plattform zu konzipieren und zu implementieren, die durch interaktive Visualisierungen von Umweltdaten gezielt zur Förderung von Data Literacy beiträgt. Dabei wurde untersucht, wie ausgewählte Komponenten der Data Literacy im schulischen Kontext mithilfe speziell entwickelter Module unterstützt werden können.

Im Rahmen der Arbeit wurden insgesamt sechs Module entwickelt: eine *Messstellen-Karte*, eine *hexagonale Dichtekarte*, eine Visualisierung des *Berliner Wärmeinseleffekts*, zwei interaktive Module zur Visualisierung von Live-Daten (*Live vor Ort* und *Live im Vergleich*) sowie ein Modul zur *interaktiven Datenexploration*. Alle Module wurden unter der Berücksichtigung der Kriterien zur Visualisierung von Umweltdaten (siehe Tabelle 3) und der typischen Herausforderungen im Umgang mit Daten (siehe Abschnitt 2.1) realisiert, die zuvor aus der Literatur abgeleitet wurden.

Die Arbeit folgte dabei einem interdisziplinären Ansatz, bei dem Konzepte und Methoden aus der Informatik mit Überlegungen der naturwissenschaftlichen Fachdidaktik verknüpft wurden. Während die technische Umsetzung auf Webtechnologien und interaktiv nutzbaren Visualisierungstools basiert, orientierte sich die didaktische Gestaltung an konkreten Anforderungen zur Förderung von Data Literacy im schulischen Kontext. Die einzelnen Module des DataLabs wurden nicht nur funktional, sondern auch mit Blick auf ihre lernförderliche Wirkung konzipiert. Dabei spielte die Frage eine zentrale Rolle, wie technische

Möglichkeiten genutzt werden können, um Lernenden den Zugang zur Arbeit mit Daten, zur kritischen Reflexion über Datenquellen und zur Interpretation komplexer Zusammenhänge in der Umwelt zu erleichtern.

Im Folgenden werden die entwickelten Module in Hinblick auf ihren Beitrag zur Erreichung dieses Ziels diskutiert. Dabei werden sowohl die Umsetzung der zuvor definierten Gestaltungsprinzipien als auch die Grenzen und Herausforderungen reflektiert, die sich aus der Verbindung von technischer Entwicklung und didaktischer Zielsetzung ergeben. Es ist jedoch hervorzuheben, dass eine empirische Evaluation zur Überprüfung des tatsächlichen didaktischen Nutzens des DataLabs nicht Bestandteil dieser Arbeit war und daher in zukünftigen Studien erfolgen sollte.

5.1 Module des DataLabs

5.1.1 Messstellen-Karte

Das Modul *Messstellen-Karte* wurde mit dem Ziel entwickelt Umweltdaten zu visualisieren. Dies geschah in Hinblick auf die Kriterien *Repräsentationsvielfalt* (Lachmayer et al., 2007; Ridsdale et al., 2015), *Interpretierbarkeit* (Lachmayer et al., 2007; Wolff et al., 2016), *Kontextualisierung* (Gott & Duggan, 2007; Hug & McNeill, 2008), *Sichtbarmachung der Datenqualität* (Gott & Duggan, 2007; Wolff et al., 2016), *mehrdimensionale Darstellung* (Hug & McNeill, 2008; Wolff et al., 2016) sowie *Ästhetik und Benutzerfreundlichkeit* (Ridsdale et al., 2015) anhand der Tabelle 3 *Kriterien zur Visualisierung von Umweltdaten im Bildungskontext Data Literacy* in Abschnitt 2.2.

Durch die Darstellung der Sensorstandorte auf einer Karte kann Lernenden ein direkter Bezug zur räumlichen Herkunft der Daten ermöglicht werden, wodurch die Kompetenz *Verständnis der Datenherkunft und Kontextualisierung* angesprochen werden kann. Auf diese Weise könnte nicht nur das Verständnis darüber gefördert werden, wo Daten erhoben werden, es ließen sich auch Rückschlüsse auf die jeweiligen Umgebungsbedingungen ziehen, unter denen die Messungen durchgeführt wurden (Hug & McNeill, 2008; Wolff et al., 2016).

Mit dem Modul soll zudem die Kompetenz *Interpretation von Zeitreihen und Diagrammen* gestärkt werden, mit dem Ziel die Fähigkeiten zum Ablesen von Diagrammen, zum Vergleichen von Messwerten und zum Erkennen von Mustern in Daten zu fördern (Lachmayer et al., 2007; Ridsdale et al., 2015). Die Kompetenz *Umgang mit Messunsicherheiten und Messlücken* sowie die Kompetenz der *Evaluation und Sicherstellung der Qualität von Daten und ihren Quellen* kann durch die Darstellung von Messlücken sowie durch die Präsentation von Messdaten in Form von Linien- und Streudiagrammen unterstützt werden (Masnick & Klahr, 2003). Die Herkunft der Second-Hand Daten und insbesondere die Messfehler darin, können dadurch besser kritisch hinterfragt werden (Hug & McNeill,

2008). Durch die visuelle Hervorhebung von Schwellwertüberschreitungen gesetzlicher Grenzwerte wird die gesellschaftliche Relevanz der Daten verdeutlicht und eine ethische Bewertung der Messreihen ermöglicht, was zur Förderung der *ethischen Kompetenz* beitragen kann (Wolff et al., 2016). Lernende werden zudem befähigt, eigenständig Hypothesen bilden und prüfen zu können. Die interaktive Darstellung von Daten, welche insbesondere durch die Möglichkeit entsteht, Graphen vergrößern zu können und die exakten Zeitpunkte der Messung einzublenden, ermöglicht eine aktive und vertiefende Exploration der Daten. Dies unterstützt die Förderung der Kompetenz *Interpretation von Daten*, da Lernende zur selbstständigen Analyse und Reflexion angeregt werden können (Lachmayer et al., 2007). Die besondere Stärke dieses Moduls liegt darin, auch komplexe Umweltinformationen ästhetisch und nutzerfreundlich aufzubereiten. Die Integration von Satellitenbildern und 3D-Gebäudemodellen tragen zur Anschaulichkeit bei und können ein realitätsnahe Lernen fördern, wie es von Wolff et al. (2016) gefordert wird.

Eine Herausforderung könnte darin bestehen, dass die Lernenden die Darstellung der Daten in Linien- und Streudiagrammen nicht eigenständig verstehen können. Besonders jüngere Lernende oder solche mit einer bisher nur gering ausgeprägten Data Literacy müssten in dem Fall eine geeignete didaktische Begleitung durch Lehrkräfte bekommen. Zudem kann die Präsentation der Daten in Form dieses Moduls nicht allein für sich stehen, wenn die Lernenden nicht in der Lage sind, eigene Forschungsfragen und Hypothesen zu bilden.

5.1.2 Hexagonale Dichtekarte

Bei dem Modul *Hexagonale Dichtekarte* standen die Kriterien *Datenqualität sichtbar machen* (Gott & Duggan, 2007; Wolff et al., 2016), *Interpretierbarkeit* (Lachmayer et al., 2007; Wolff et al., 2016) und *Kontextualisierung* (Gott & Duggan, 2007; Hug & McNeill, 2008) im Vordergrund.

Die Wahl fiel auf die Darstellung der Daten in Form von hexagonalen Gebieten auf einer Landkarte. Bei dieser Art der Darstellung werden die Messdaten nicht auf konkrete geografische oder „künstlich“ definierte Gebiete bezogen. Stattdessen werden die Daten homogen gerastert auf einer Landkarte verteilt dargestellt. Dies verhindert eine verzerrte Bewertung, die allein auf der unterschiedlichen Größe von durch den Menschen definierten Gebieten beruht. Jede Fläche in dieser Darstellung ist gleich groß und es findet keine selektive Gewichtung nach der Lage des Sensors statt. Die Lernenden werden so in die Lage versetzt, die Daten ohne die übermäßige Verzerrung von geografischen Grenzen zu betrachten, wie sie bei Flächenkartogrammen auftritt. Die Kontextualisierung der Daten zu ihrem Standort wird hergestellt, indem die Messstationen trotzdem mit auf der Karte eingezeichnet werden.

Die Qualität der Daten wird sichtbar gemacht, da nur die obersten und untersten 1 % der Daten verworfen werden. Bei dieser leichten Form der Datenbearbeitung werden die ex-

tremen Messwerte entfernt, welche die Darstellung andernfalls so stark verzerrten würden, dass eine sinnvolle Einteilung der Skala in der Legende nicht mehr möglich wäre. Alle anderen Daten, auch sonstige Messfehler, werden bewusst unverarbeitet angezeigt. Das ermöglicht es den Lernenden, sich kritisch mit den Daten und den Ursachen für diese Darstellung der Werte auseinanderzusetzen.

Die Erreichung der didaktischen Ziele wurde angestrebt, indem die Karte zusätzlich eine räumlich-zeitliche Darstellung anbietet. Die Daten lassen sich in statischer oder animierter Form darstellen, was zu einer explorativen Analyse ermutigen soll. Das wäre nicht möglich, wenn nur einfache Linien-, Balken- oder Streudiagramme verwendet werden würden.

Durch die Kontextualisierung und die Verortung der Messtationen auf einer Karte kann die Kompetenz *Sammeln von Daten* gefördert werden, da das Verständnis über den Entstehungskontext von Daten, hier ein wesentliches Kernkonzept darstellt (Hug & McNeill, 2008; Masnick & Klahr, 2003; Ridsdale et al., 2015).

Die Stärke dieses Moduls liegt in der abstrakten und zugleich ansprechenden Darstellung von Daten, die durch die Animation der Zeitreihen eine dynamische und tiefergehende Exploration der räumlichen und zeitlichen Zusammenhänge ermöglicht. Die komplexe Form der Datenvisualisierung, die von traditionellen Darstellungsformen abweicht, kann das *analytische Denken* fördern und die Lernenden zur Entwicklung eigener Hypothesen animieren (Hug & McNeill, 2008; Ridsdale et al., 2015; Wolff et al., 2016).

Ob das Modul das Versprechen einhält, die erwähnten Kriterien umzusetzen und die Kompetenzen zu fördern, muss letztlich eine separate Evaluation prüfen. Das Modul lässt sich durch die Bereitstellung von Begleittexten auf der Webseite sowie einblendbare Hinweisfelder beliebig auf die Ziele von Lerneinheiten hin optimieren.

5.1.3 Berliner Wärmeinseleffekt

Das Modul *Berliner Wärmeinseleffekt* wurde geschaffen, um die Kriterien *Interpretierbarkeit* (Lachmayer et al., 2007; Wolff et al., 2016), *Kontextualisierung* (Gott & Duggan, 2007; Hug & McNeill, 2008) und *mehrdimensionale Darstellung* (Hug & McNeill, 2008; Wolff et al., 2016) für die Visualisierung von Umweltdaten umzusetzen. Die gemischte Darstellung von zwei Werten, der Lufttemperatur und der Ausprägung von Wind, in Form von bivariaten Flächenkartogrammen ermöglicht eine Visualisierung komplexer Umeldaten, die zugleich intuitiv verstehbar ist.

Die Erreichung der didaktischen Ziele dieses Modul wurden angestrebt, indem Lernende in die Lage versetzt werden, die komplexen räumlichen und zeitlichen Zusammenhänge von Temperatur und Windverhältnissen innerhalb der städtischen Struktur in einer Grafik vereint zu sehen. Dadurch können die Kompetenzen *Interpretation von Daten* und *Identifizierung von Problemen mittels Daten* gefördert werden (Lachmayer et al., 2007; Ridsdale

et al., 2015; Wolff et al., 2016). Durch die zusätzliche Möglichkeit die Daten in einer animierten Form über den Tagesverlauf hinweg anzuzeigen, werden Lernende in die Lage versetzt die Auswirkung von Umwelteinflüssen in einer zeitlichen Dynamik zu analysieren und zu interpretieren, was die Kompetenz der *datengestützten Entscheidungsfindung* fördern kann (Hug & McNeill, 2008; McNeill & Krajcik, 2007; Schable et al., 1995; Wolff et al., 2016).

Der Berliner Wärmeinseleffekt wird den Lernenden unmittelbar verständlich gemacht und kann zu einer vertiefenden Auseinandersetzung mit den zu Grunde liegenden Phänomenen motivieren, worin die Stärke des Moduls gesehen werden kann. Lernende sollen angeregt werden eigene Hypothesen über die Ursachen der Entstehung von Wärmeinseln zu entwickeln. Zudem sollen sie die Hypothesen hinterfragen, indem sie die visualisierten Datensätzen kritisch bewerten. Die komplexe Visualisierungsform kann die *analytischen Fähigkeiten* und das *kritische Denken* fördern, wie es für eine moderne Data Literacy wesentlich ist (Ridsdale et al., 2015; Wolff et al., 2016).

Für die Lernenden kann die Komplexität der Darstellung herausfordernd sein, obwohl versucht wurde, eine Balance zwischen der Menge an dargestellten Informationen und einer guten Verständlichkeit der Grafiken zu wahren. Insbesondere bei einer jüngeren oder einer weniger kompetenten Zielgruppe, erscheint eine didaktische Begleitung durch Lehrkräfte notwendig. Da es darum geht, den Zusammenhang zwischen dem Versiegelungsanteil einerseits und der Temperatur und dem Vorhandensein von Wind andererseits herzustellen, kann eine Hilfestellung bei der Erstellung von Forschungsfragen und Hypothesen durch zielgerichtete Fragen zu dem Effekt seitens der Lehrkräfte erfolgen.

5.1.4 Live vor Ort – Sensordaten am Standort

Mit dem Modul *Live vor Ort – Sensordaten am Standort* wurden die Kriterien *Repräsentationsvielfalt* (Lachmayer et al., 2007; Ridsdale et al., 2015), *Interpretierbarkeit* (Lachmayer et al., 2007; Wolff et al., 2016), *Kontextualisierung* (Gott & Duggan, 2007; Hug & McNeill, 2008), *Handlungsrelevanz* (Ridsdale et al., 2015; Wolff et al., 2016), *Bezug zu eigenen Daten* (Hug & McNeill, 2008; Wolff et al., 2016), *Datenqualität sichtbar machen* (Gott & Duggan, 2007; Wolff et al., 2016), *mehrdimensionale Darstellung* (Hug & McNeill, 2008; Wolff et al., 2016) sowie *Ästhetik und Benutzerfreundlichkeit* (Ridsdale et al., 2015) für die Visualisierung von Umweltdaten umgesetzt.

Die didaktischen Ziele sollen erreicht werden, indem Lernende in die Lage versetzt werden mittels den ausgegebenen SenseBoxen selbstständig Messdaten zu erfassen und durch die direkte Anzeige der Werte, eigene Forschungen anstellen zu können. Die Messwerte können entweder zusammengefasst für alle SenseBoxen mit demselben GroupTag angezeigt werden oder (nach Auswahl über die obere Menüleiste) gezielt für eine einzelne SenseBox. Die Lernenden können so nicht nur einen direkten Bezug zu dem Kontext herstellen,

in welchem die Daten erhoben wurden, sie erfahren auch, welche Faktoren die Messungen beeinflussen und verfälschen können. Die Standortkarte, die oberhalb der Kacheln mit den Sensorwerten angezeigt wird, fügt den Daten weiteren Kontext hinzu. Neben dem Standort der SenseBox können in der Karte zusätzliche Ebenen wie eine Satellitenansicht aktiviert werden, die eine Ansicht des eigenen Standorts aus der Vogelperspektive bietet. Einblendbare 3D-Gebäude ermöglichen, den Einfluss von Bebauung auf die Messdaten zu bewerten.

Eine Stärke des Moduls besteht darin, dass Lernende mit authentischen und selbst erhobenen Daten arbeiten können, was die Kompetenzen *Datenermittlung und -erfassung* sowie *Evaluation und Sicherstellung der Qualität von Daten und ihren Quellen* fördern kann (Hug & McNeill, 2008; Masnick & Klahr, 2003; McNeill & Krajcik, 2007; Wolff et al., 2016). Die Lernenden werden in die Lage versetzt Unstimmigkeiten in den Daten zu identifizieren und diese mit weiteren relevanten Informationen (etwa den Umgebungsbedingungen) zu verknüpfen, was die Kompetenz der *Interpretation von Daten* fördern kann (Lachmayer et al., 2007; Wolff et al., 2016). Durch die interaktive Darstellung von Messwerten in Echtzeit kann die Kompetenz *Präsentation von Daten* unterstützt werden, da Lernende die Visualisierungen zur gemeinsamen Analyse und Diskussion in der Gruppe heranziehen können (Gott & Duggan, 2007; Lachmayer et al., 2007; Newton et al., 1999).

Eine zentrale Herausforderung bestand darin, die Daten der openSenseMap-API in kurzer zeitlicher Abfolge abzurufen, um den Lernenden möglichst aktuelle Messwerte bereitzustellen zu können. Wie in Abschnitt 3.4.4 beschrieben, sind hierfür mindestens zwei aufeinanderfolgende API-Abfragen erforderlich, um vollständige Sensordaten zu erhalten. Zu Beginn jeder Messreihe ist darüber hinaus eine weitere Abfrage nötig, um die zugehörigen Messeinheiten zu ermitteln. Die erhaltenen Daten müssen anschließend mit erheblichem Aufwand in ein einheitliches Format überführt und zusammengeführt werden, um sie gemeinsam in einer Darstellung visualisieren zu können.

Dieser Prozess erwies sich zum einen als technisch aufwendig, zum anderen als performancekritisch, da jede zusätzliche Abfrage die lokale und externe Infrastruktur belastet und spürbar zur Antwortzeit beiträgt. Besonders erschwerend kommt hinzu, dass die API bei vielen Anfragen vergleichsweise langsam reagiert und keine Unterstützung für Websockets bietet, wodurch eine Echtzeitübertragung im engeren Sinne nicht möglich ist. Beide Problembereiche sind bereits durch Bugreports dokumentiert („Migrate to Postgres · Issue #846“, 2023; „Real time · Issue #40“, 2016).

Darüber hinaus zeigte die API in der praktischen Umsetzung teils inkonsistentes und schwer vorhersehbares Verhalten, was die Implementierung robuster und fehlertoleranter Abrufmechanismen erforderlich machte. Rückblickend beanspruchte die Integration der API einen erheblichen Anteil der gesamten Entwicklungszeit und stellte damit einen wesentlichen limitierenden Faktor bei der Umsetzung dar.

Herausfordernd aus Perspektive der Lernenden könnten sein, dass nicht intuitiv klar ist, was mit den GroupTags gemeint ist und welche Funktion sie im DataLab erfüllen. Auch die Darstellung der SenseBoxen, ob einzeln oder zusammengefasst, kann herausfordernd wirken, da viele Informationen gleichzeitig präsentiert werden. Zwar wurde bei der visuellen Gestaltung dieses Moduls darauf geachtet, dass die Darstellung nicht nur visuell ansprechend ist, sondern die Einteilung der Benutzeroberfläche auch einem festen Schema folgt (unter der Bezeichnung für die SenseBoxen befindet sich eine Karte mit dem Standort; darunter folgen die Darstellungen für die Sensoren; diese sind in Kacheln mit Piktogrammen, einer Beschriftung und den Diagrammen organisiert). Eine didaktische Begleitung durch eine Lehrkraft kann an diesen Stellen dennoch von Nöten sein, um eine Überforderung der Lernenden zu verhindern.

Kritisch kann auch betrachtet werden, dass als mittlerer Wert bzw. Zentralwert für die Messwerte mehrerer SenseBoxen der Median herangezogen wurde, was mathematisch zwar plausibel ist (da der Median robuster gegenüber Ausreißern ist als das arithmetische Mittel), für Lernende der unteren Klassenstufen das Konzept des Medians jedoch nicht vertraut sein dürfte.

5.1.5 Live im Vergleich – Sensordaten über Standorte hinweg

Das Modul *Live im Vergleich – Sensordaten über Standorte hinweg* wurde entwickelt, um mehrere Kriterien zur Visualisierung von Umweltdaten umzusetzen, insbesondere *Repräsentationsvielfalt* (Lachmayer et al., 2007; Ridsdale et al., 2015), *Interpretierbarkeit* (Lachmayer et al., 2007; Wolff et al., 2016), *Kontextualisierung* (Gott & Duggan, 2007; Hug & McNeill, 2008), *Handlungsrelevanz* (Ridsdale et al., 2015; Wolff et al., 2016), *Interaktivität/Exploration* (Ridsdale et al., 2015; Wolff et al., 2016), *Bezug zu eigenen Daten* (Hug & McNeill, 2008; Wolff et al., 2016), *Vergleich und Mustererkennung* (Hug & McNeill, 2008; Lachmayer et al., 2007), *Sichtbarmachung der Datenqualität* (Gott & Duggan, 2007; Wolff et al., 2016), *mehrdimensionale Darstellung* (Hug & McNeill, 2008; Wolff et al., 2016) sowie *Ästhetik und Benutzerfreundlichkeit* (Ridsdale et al., 2015).

Die didaktischen Ziele des Moduls sollen erreicht werden, indem Lernenden eine interaktive Umgebung geboten wird, innerhalb derer sie die Echtzeitdaten von mehreren Standorten gleichzeitig betrachten und vergleichen können. Sie bekommen hier die Gelegenheit Einflüsse von Umweltbedingungen (wie etwa Flächenversiegelung, Einfluss von Vegetation, Entfernung vom Stadtzentrum, ...) auf die Messwerte zu erfassen und mit eigenen oder vorgegebenen Forschungsfragen kritisch zu bewerten. Die flexible Auswahl von SenseBoxen ermöglicht es die Oberfläche individuell auf spezifische Fragestellungen und Unterrichtsszenarien anzupassen. Dadurch können die Kompetenzen *Datenermittlung und -erfassung*, *Evaluation und Sicherstellung der Qualität von Daten und ihren Quellen* sowie die *Interpretation von Daten* gefördert werden. Durch den Vergleich von Messwerten unterschiedlicher Standorte werden Lernende dazu angeregt, Muster in den Daten zu identifizieren und

sich kritisch mit den Ursachen möglicher Abweichungen auseinanderzusetzen. Die Darstellung der Messwerte in nahezu Echtzeit erhöht die Authentizität der Lernerfahrung und kann die Lernenden motivieren, eigenständig komplexe Analysen durchzuführen (Hug & McNeill, 2008; Lachmayer et al., 2007; Wolff et al., 2016).

Herausforderungen können sich dadurch ergeben, dass die Benutzeroberfläche auch schnell überladen wirken kann, wenn sehr viele Standorte gleichzeitig angezeigt werden. Die Übersichtlichkeit kann insbesondere dann verloren gehen, wenn mehr als vier Standorte gleichzeitig zur Ansicht ausgewählt werden.

Die in *Live vor Ort – Sensordaten am Standort* (Abschnitt 5.1.4) erwähnten Probleme treffen auch in diesem Modul zu, was zu Verzögerungen in der Darstellung von Daten führen kann.

Eine didaktische Anleitung kann in diesem Modul insbesondere dann sinnvoll sein, wenn spezifische Forschungsfragen bearbeitet und geeignete Standorte und Sensoren ausgewählt werden sollen.

5.1.6 Interaktive Datenexploration

Das Modul *Interaktive Datenexploration* wurde entwickelt, um spezifisch die Kriterien *Interpretierbarkeit* (Lachmayer et al., 2007; Wolff et al., 2016), *Kontextualisierung* (Gott & Duggan, 2007; Hug & McNeill, 2008), *Interaktivität/Exploration* (Ridsdale et al., 2015; Wolff et al., 2016) sowie *Adaptierbarkeit* (Wolff et al., 2016) bei der Visualisierung von Umweltdaten umzusetzen. Durch dieses Modul erhalten Lernende die Möglichkeit eigenständige, interaktive und individuell konfigurierte Diagramme für die Visualisierung von Umweltdaten zu erstellen.

Didaktische Ziele sollen erreicht werden, indem Lernende befähigt werden eigene *Visualisierungen von Umweltdaten* vor dem Hintergrund einer selbst gestellten oder vorgegebenen Forschungsfrage zu entwerfen und diese auf ihre Wirksamkeit hin zu beurteilen (McDermott et al., 1987; Wolff et al., 2016). Sie bekommen hier die Möglichkeit alle Parameter (Sensortyp, Zeitfenster, Farbskala, Kartentyp, Größe der hexagonalen Flächen), die das Aussehen der hexagonalen Dichtekarte beeinflussen, selbst zu wählen. Diese Möglichkeit zur Bearbeitung bzw. Manipulation soll sie in die Lage versetzen, kritisch zu bewerten welcher Zusammenhang zwischen der Darstellung von Daten und der inhaltlichen Interpretation besteht. Es besteht die Möglichkeit, eine kritische Beurteilung der Genauigkeit und Eindeutigkeit grafischer Auswertungen zu üben. Zudem können Lernende sich in der Bewertung üben, inwieweit die generierten grafischen Darstellungen die erhobenen Daten in geeigneter Weise repräsentieren.

Durch diese Vorgehensweise könnten die Kompetenzen *Interpretation von Daten*, *Evaluation von Darstellungen* und *Präsentieren von Daten* gefördert werden (Gott & Duggan,

2007; Hug & McNeill, 2008; Lachmayer et al., 2007).

Dieses Modul ist nicht wie die anderen Module in der Seitenleiste des DataLabs eingebettet, sondern verbirgt sich unter der Bezeichnung „Eigene Diagramme erstellen“ etwas versteckt in dem Zusatzmenü, dass sich hinter dem Zahnrad Symbol unten links befindet, da es nicht für die jüngste Zielgruppe geeignet erscheint. Aufgrund der Vielzahl an Konfigurationsmöglichkeiten und der Offenheit in der Gestaltung der Visualisierungen könnte die Einstiegshürde für Lernende ohne ausreichende Vorerfahrungen zu hoch sein. Dies könnte insbesondere bei jüngeren oder weniger datenkompetenten Lernenden zu Orientierungsschwierigkeiten führen. Dieser Herausforderung könnte künftig begegnet werden, indem zusätzliche unterstützende Ressourcen wie Schritt-für-Schritt-Anleitungen, didaktische Hinweise oder eingebettete Tutorials bereitgestellt werden, um die Lernenden strukturiert an diese explorative Arbeitsweise heranzuführen.

5.2 Technische Realisation

Das DataLab als integrierte Plattform zu Visualisierung von Umweltdaten im Kontext Data Literacy wurde als ein modulares System geplant und umgesetzt. Hierzu wurde ein bekanntes, stabiles und weit verbreitetes Webframework verwendet, welches schrittweise um Funktionen erweitert wurde. Dazu gehörte unter anderem die Einbettung eines endanwenderfreundlichen CMS, um die Erstellung und Verwaltung von zusätzlichen (didaktischen) Contents möglich zu machen. In dieses System wurde dann die Funktionalität mit `collect_data.py` eingebaut, um die openSenseMap-API möglichst schnell und robust abrufen zu können. Die abgerufen Daten werden kontextabhängig in drei jeweils für ihren Einsatzzweck geeigneten Datenbanken gespeichert. In dieses System wurden dann die einzelnen Module integriert.

Das Design des DataLabs bietet nach dem Entwurf und der umfangreichen Erprobung eine robuste Grundlage, um beliebig weitere Module mit anderen Schwerpunkten zu integrieren. Die Wahl eines containerbasierten Ansatzes für das DataLab erlaubt eine einfache und schnelle Installation in einer neuen Umgebung.

Das DataLab setzt eine 64-bit fähige CPU, mindestens 8 GB Hauptspeicher, mindestens zwei CPU-Kerne und einen verfügbaren Festplattenplatz von 50 GB voraus. Es ist möglich die Abfrage auf weitere APIs zu erweitern, die Ressourcen sollten auch dafür reichen. Das Design der Funktion `collect_data.py` lässt sich hierfür durch den universellen Ansatz, der hier verwendet wurde, schnell erweitern. Die Performance kann zukünftig nach Bedarf weiter gesteigert werden, indem mehr Funktionen an die asynchrone Task Queue Celery ausgelagert werden, wodurch aber auch der Ressourcenbedarf des DataLabs weiter ansteigen würde.

Der Code hält sich an gängige Standards, das Framework ist weit verbreitet, die Program-

miersprache ist im wissenschaftlichen Kontext (vor allem in der Data Science) geläufig und der Code ist reich kommentiert, modularisiert und verwendet an vielen Stellen Annotationen, was günstig für die zukünftige Stabilität, Pflege und Erweiterbarkeit ist.

5.3 Anwendungsfreundlichkeit im Unterricht

Es sollte geprüft werden, ob die Darstellung der Diagramme für die Altersgruppe der Lernenden verständlich genug ist und ob die Bedienung soweit intuitiv möglich ist, dass die Lernerfahrung nicht gestört wird. Zudem sollte untersucht werden, welche Darstellungen für Unterrichtseinheiten geeignet sind und ob den implementierten Modulen noch wichtige Elemente für das Verständnis fehlen.

Die Wahl der Symbole in der Seitenleiste kann noch optimiert werden, indem selbsterklärende oder optisch ansprechendere und ggf. größere Piktogramme für die Zielgruppe ausgewählt werden.

Aus technischen Gründen kommt es immer wieder zu Wartezeiten, die sich nur begrenzt beheben lassen. An diesen Stellen könnten noch weitere Ladeanimationen anzeigen, dass das DataLab mit der Bearbeitung der Anfrage beschäftigt ist. Es ist davon auszugehen, dass das DataLab auch Lastsituationen aushalten kann, wie sie in Lerneinheiten mit mehreren dutzend Geräten und vielen Anfragen pro Minute entstehen. Eine Situation mit dieser hohen Belastung ist aber noch nicht eingetreten und konnte daher nicht geprüft werden.

Bei dem Entwurf des DataLabs wurde immer mitgedacht, wie eine typische Unterrichtssituation aussieht: Lernende greifen über einen mobilen Hotspot mit limitierten Übertragungsvolumen und eingeschränkter Geschwindigkeit mit iPads auf die Seite des DataLabs zu. Dabei ist wichtig, dass die Darstellung nicht nur auf dieser Displaygröße gut funktioniert, die Seite soll mit möglichst geringen Datenmengen auskommen und den Endgeräten an möglichst vielen Stellen erlauben die Ergebnisse (Bilder, Diagramme, Symbole, Animationen, ...) zwischenzuspeichern, um ein erneutes Übertragen des selben Inhalts zu vermeiden.

Das DataLab hat die Grundlagen eingebaut, um mehrsprachig zu sein. Dafür können die im Abschnitt 3.2.2 Struktur des DataLabs erwähnten Textdateien mit der Endung `.po` bearbeitet werden. Es lassen sich neben dem Standort Berlin auch weitere Standorte zur Visualisierung von Umweltdaten hinzufügen (siehe Abschnitt 3.3 Abruf von Umweltdaten).

6 Schlussfolgerung und Ausblick

Die vorliegenden Ergebnisse legen nahe, dass die entwickelte Plattform grundsätzlich dazu geeignet ist, wichtige Teilkompetenzen der Data Literacy zu fördern. Im Vordergrund

stehen hier die Kompetenzen der Datenerfassung, Datenanalyse, Dateninterpretation sowie der kritische Umgang mit Datenquellen und Datenqualität. Die konzipierten Module ermöglichen dabei eine zielgerichtete und abwechslungsreiche Vermittlung dieser Kompetenzen, indem sie reale Umweltdaten (sowohl in Echtzeit als auch in aufgezeichneter Form) in einen lebensnahen und unmittelbar relevanten Kontext setzen.

Bei der Entwicklung des DataLabs wurden Erkenntnisse aus der Theorie und Empirie in Hinblick auf Data Literacy berücksichtigt. Es wurde angestrebt, spezifisch auf die Anforderungen im Rahmen des Projekts der *Humboldt Explorers* einzugehen und zugleich die Plattform so offen und anpassungsfähig zu gestalten, dass sie auch im Rahmen anderer Projekte und für andere Zielgruppen verwendet werden kann. Der Schwerpunkt der vorliegenden Arbeit liegt in der technischen Realisation dieses theoriegeleiteten Vorgehens.

Im weiteren Verlauf müssen nun die Module in didaktisch aufbereitete Lerneinheiten eingepflegt werden sowie alle Begleittexte auf der Webseite formuliert werden, die eine zielgruppengerechte Einführung gewährleisten können, so dass sich die wesentlichen Zusammenhänge für Lehrende und Lernende erschließen.

Der modulare Aufbau des DataLabs kann sowohl in Seminaren von Lehramtsstudierenden der MINT-Fächer als auch in Lehrkräftefortbildungen genutzt werden, um Unterrichtsminiaturen und begleitende Interventionen zu konzipieren.

Notwendig ist dann die praktische Erprobung des DataLabs durch Lernende und Lehrkräfte. In diesem Zusammenhang ist auch die Nutzerführung und Usability zu testen. Die eigentliche Evaluation des didaktischen Nutzens des DataLabs muss anschließend in empirischen Studien erfolgen.

Die gewonnenen Erkenntnisse können dann wiederum zu künftigen Anpassungen und Erweiterungen des DataLabs führen. Denkbar wäre etwa die Erweiterung um Module, welche auf die Bestimmung pflanzlicher Artenvielfalt auf Berlins Schulhöfen oder auf die Quantifizierung tropischer Nächte abzielen. Auch ein internationaler Vergleich von Messdaten aus Großstädten unterschiedlichster Länder wäre möglich.

Interessant wäre überdies die Einbindung von weiteren externen Quellen wie beispielsweise den Energy-Charts (Burger, 2025). Hier könnte etwa ein Modul entwickelt werden, was sich mit Stromproduktion, Stromerzeugung, Emissionen, Klimadaten, Spotmarktpreisen und Szenarien zur Energiewende beschäftigt.

In der Möglichkeit zur stetigen Anpassung, Weiterentwicklung und Verbesserung des DataLabs kann dessen besondere Stärke gesehen werden.

Literaturverzeichnis

- Börner, K., Bueckle, A., & Ginda, M. (2019). Data visualization literacy: Definitions, conceptual frameworks, exercises, and assessments. *Proceedings of the National Academy of Sciences*, 116(6), 1857–1864. <https://doi.org/10.1073/pnas.1807180116>
- Burger, B. (2025, 18. Mai). *Energy-Charts; Fraunhofer-Institut für Solare Energiesysteme ISE*. Verfügbar 18. Mai 2025 unter <https://www.energy-charts.info/>
- Carr, D. B., Littlefield, R. J., Nicholson, W. L., & Littlefield, J. S. (1987). Scatterplot Matrix Techniques for Large N. *Journal of the American Statistical Association*, 82(398), 424–436. <https://doi.org/10.1080/01621459.1987.10478445>
- Continuous Color Scales in Python Plotly*. (2025). Verfügbar 25. März 2025 unter <https://plotly.com/python/builtin-colorscales/>
- Deahl, E. (2014, 9. Mai). Better the data you know: Developing youth data literacy in schools and informal learning environments. <https://doi.org/10.2139/ssrn.2445621>
- Django Software Foundation. (2024). *Django* (Version 5.1.4). Verfügbar 18. Dezember 2024 unter <https://www.djangoproject.com/>
- Django Software Foundation. (2025). *Django template language* (Version 5.1.4). Verfügbar 18. Dezember 2024 unter <https://docs.djangoproject.com/en/5.1/ref/templates/language/>
- FIS-Broker: Lufttemperatur Daten*. (2025). Verfügbar 28. März 2025 unter https://fbinter.stadt-berlin.de/fb/index.jsp?loginkey=showMap&mapId=wmsk_04102_lufttemp2015@senstadt
- FIS-Broker: Winddaten*. (2025). Verfügbar 28. März 2025 unter https://fbinter.stadt-berlin.de/fb/index.jsp?loginkey=showMap&mapId=wmsk_04101_windf2015@senstadt
- Garcia-Mila, M., Pérez-Echeverría, M. P., Postigo, Y., Martí, E., Villarroel, C., & Gabucio, F. (2016). Nuclear power plants? Yes or no? Thank you! The argumentative use of tables and graphs / ¿Centrales nucleares? Sí o no? ¡Gracias! El uso argumentativo de tablas y gráficas. *Journal for the Study of Education and Development*, 39(1), 187–218. <https://doi.org/10.1080/02103702.2015.1111605>
- Gee, J. P. (1996). *Social Linguistics and Literacies: Ideology in Discourses* (2. Aufl.). Falmer Press.
- Germann, P. J., & Aram, R. J. (1996). Student performances on the science processes of recording data, analyzing data, drawing conclusions, and providing evidence. *Journal of Research in Science Teaching*, 33(7), 773–798. [https://doi.org/10.1002/\(SICI\)1098-2736\(199609\)33:7<773::AID-TEA5>3.0.CO;2-K](https://doi.org/10.1002/(SICI)1098-2736(199609)33:7<773::AID-TEA5>3.0.CO;2-K)
- Gott, R., & Duggan, S. (2007). A framework for practical work in science and scientific literacy through argumentation. *Research in Science & Technological Education*, 25(3), 271–291. <https://doi.org/10.1080/02635140701535000>

- Gross, C. (2024, 13. Dezember). *</> htmx - high power tools for html* (Version 1.9.12). <https://htmx.org/>
- Hogan, K., & Maglienti, M. (2001). Comparing the epistemological underpinnings of students' and scientists' reasoning about conclusions. *Journal of Research in Science Teaching*, 38(6), 663–687. <https://doi.org/10.1002/tea.1025>
- Hug, B., & McNeill, K. L. (2008). Use of first-hand and second-hand data in science: Does data type influence classroom conversations? *International Journal of Science Education*, 30(13), 1725–1751. <https://doi.org/10.1080/09500690701506945>
- Humboldt Explorers*. (2021, 29. November). Verfügbar 17. Februar 2025 unter <https://www.humboldt-explorers.de/>
- Krajcik, J., McNeill, K. L., & Reiser, B. J. (2008). Learning-goals-driven design model: Developing curriculum materials that align with national standards and incorporate project-based pedagogy. *Science Education*, 92(1), 1–32. <https://doi.org/10.1002/sce.20240>
- Kruchten, N., Seier, A., & Parmer, C. (2024, September). *plotly* (Version 5.24.1). Plotly Technologies Inc. <https://doi.org/10.5281/zenodo.14503524>
- Kultusministerkonferenz. (2016). Bildung in der digitalen Welt: Strategie der Kultusministerkonferenz.
- Lachmayer, S., Nerdel, C., & Prechtl, H. (2007). Modellierung kognitiver Fähigkeiten beim Umgang mit Diagrammen im naturwissenschaftlichen Unterricht. *Zeitschrift für Didaktik der Naturwissenschaften : ZfDN*, 13, 145–160. <https://doi.org/10.25656/01:31610>
- Lufttemperatur Umweltatlas Berlin*. (2015, August). Verfügbar 28. März 2025 unter <https://www.berlin.de/umweltatlas/klima/klimaanalyse/2014/karten/artikel.1024086.php>
- Mandinach, E. B., & Gummer, E. S. (2013). A Systemic View of Implementing Data Literacy in Educator Preparation. *Educational Researcher*, 42(1), 30–37. <https://doi.org/10.3102/0013189X12459803>
- Masnick, A. M., & Klahr, D. (2003). Error Matters: An Initial Exploration of Elementary School Children's Understanding of Experimental Error. *Journal of Cognition and Development*, 4(1), 67–98. <https://doi.org/10.1080/15248372.2003.9669683>
- Masnick, A. M., & Knowles, E. R. (2017). Data-Driven Belief Revision in Children and Adults. *Journal of Cognition and Development*, 18(1), 87–109. <https://doi.org/10.1080/15248372.2016.1168824>
- McDermott, L. C., Rosenquist, M. L., & van Zee, E. H. (1987). Student Difficulties in Connecting Graphs and Physics: Examples from Kinematics. *American Journal of Physics*, 55(6), 503–513. <https://doi.org/10.1119/1.15104>
- McNeill, K. L., & Krajcik, J. (2007). Middle school students' use of appropriate and inappropriate evidence in writing scientific explanations. *Thinking with data*, 233–265.

- Migrate to postgres · issue #846* [GitHub]. (2023, 23. Oktober). Verfügbar 22. März 2025 unter <https://github.com/sensebox/openSenseMap-API/issues/846>
- Newton, P., Driver, R., & Osborne, J. (1999). The place of argumentation in the pedagogy of school science. *International Journal of Science Education*, 21(5), 553–576. <https://doi.org/10.1080/095006999290570>
- openSenseMap.org*. (2025, 10. März). <https://opensensemap.org/>
- pandas development team, Wes McKinney. (2024, 20. September). *Pandas* (Version 2.2.3). Zenodo. <https://doi.org/10.5281/zenodo.3509134>
- Python Software Foundation, G. v. R. (2022, 24. Mai). *Python* (Version 3.12). <https://peps.python.org/pep-0693/>
- Real time · issue #40* [GitHub]. (2016, 5. Oktober). Verfügbar 22. März 2025 unter <https://github.com/sensebox/openSenseMap-API/issues/40>
- Reedu GmbH & Co. KG. (2025, 16. Januar). *senseBox*. <https://docs.opensensemap.org/>
- Richtlinie 2008/50/EG des Europäischen Parlaments und des Rates vom 21. Mai 2008 über Luftqualität und saubere Luft für Europa (2008, 21. Mai). Verfügbar 26. März 2025 unter <http://data.europa.eu/eli/dir/2008/50/oj/deu>
- Ridsdale, C., Rothwell, J., Smit, M., Bliemel, M., Irvine, D., Kelley, D., Matwin, S., Wetherick, B., & Ali-Hassan, H. (2015). Strategies and best practices for data literacy education knowledge synthesis report. <https://doi.org/10.13140/RG.2.1.1922.5044>
- Rohdaten Erfrischungskarte*. (2021, 24. Juni). Verfügbar 28. März 2025 unter https://github.com/technologiestiftung/erfrischungskarte-daten/tree/main/Wind_Temperature/data/raw
- Schauble, L., Glaser, R., Duschl, R. A., Schulze, S., & Jenny, J. (1995). Students' Understanding of the Objectives and Procedures of Experimentation in the Science Classroom. *Journal of the Learning Sciences*, 4(2), 131–166. https://doi.org/10.1207/s15327809jls0402_1
- Schnotz, W. (1992). *Wissenserwerb mit logischen Bildern*. Universität, Deutsches Institut für Fernstudien, Arbeitsbereich Lernforschung.
- Schnotz, W. (1997). Wissenserwerb mit Diagrammen und Texten. In L. J. Issing & P. Klimsa (Hrsg.), *Information und Lernen mit Multimedia*. (S. 84–105). Psychologie Verl.-Union.
- Schüller, K., Koch, H., & Rampel, F. (2021). Data-Literacy-Charta. *Data-Literacy-Charta, Version 1.2*.
- Schwippert, K., Kasper, D., Eickelmann, B., Goldhammer, F., Köller, O., Selter, C., & Steffensky, M. (Hrsg.). (2024, 4. Dezember). *TIMSS 2023. Mathematische und naturwissenschaftliche Kompetenzen von Grundschulkindern in Deutschland im internationalen Vergleich*. Waxmann Verlag GmbH. <https://doi.org/10.31244/9783830999591>

- Senatsverwaltung für Stadtentwicklung, Bauen und Wohnen. (2015, Juni). *Umweltatlas, Klimamodell Berlin - Analysekarten 2014*. Verfügbar 26. März 2025 unter <https://www.berlin.de/umweltatlas/klima/klimaanalyse/2014/karten/>
- Smith, D., Myers, J. S., Kaplan, C. S., & Goodman-Strauss, C. (2024). An aperiodic monotile. *Combinatorial Theory*, 4(1). <https://doi.org/10.5070/C64163843>
- Swan, K., Vahey, P., Kratcoski, A., van 't Hooft, M., Rafanan, K., & Stanford, T. (2009). Challenges to Cross-Disciplinary Curricula: Data Literacy and Divergent Disciplinary Perspectives. *Annual Conference of the American Educational Research Association (AERA)*.
- Technologie Stiftung Berlin. (2023). *Erfrischungskarte der Technologie Stiftung Berlin* [Technologiestiftung Berlin]. Verfügbar 26. März 2025 unter <https://erfrischungskarte.odis-berlin.de/>
- Torchbox. (2024). *Wagtail* (Version 6.3.1). Verfügbar 18. Dezember 2024 unter <https://wagtail.org/>
- Vahey, P., Yarnall, L., Patton, C., Zalles, D., & Swan, K. (2006). Mathematizing Middle School: Results From a Cross-Disciplinary Study of Data Literacy. *Proceedings of the Annual Meeting of the American Educational Research Association (AERA)*. <https://www.sri.com/publication/mathematizing-middle-school-results-from-a-cross-disciplinary-study-of-data-literacy/>
- Warren, B., Ballenger, C., Ogonowski, M., Rosebery, A. S., & Hudicourt-Barnes, J. (2001). Rethinking diversity in learning science: The logic of everyday sense-making. *Journal of Research in Science Teaching*, 38(5), 529–552. <https://doi.org/10.1002/tea.1017>
- Wilke, S. (2015, 25. Juni). *Indikator: Belastung der Bevölkerung durch Feinstaub (PM2,5)* [Umweltbundesamt]. Verfügbar 26. März 2025 unter <https://www.umweltbundesamt.de/daten/umweltindikatoren/indikator-belastung-der-bevoelkerung-durch-0>
- Wolff, A., Gooch, D., Cavero Montaner, J. J., Rashid, U., & Kortuem, G. (2016). Creating an understanding of data literacy for a data-driven society. *The Journal of Community Informatics*, 12(3). <https://doi.org/10.15353/joci.v12i3.3275>

Danksagung

Diese Arbeit wurde in der Arbeitsgruppe „Fachdidaktik und Lehr-/Lernforschung Biologie“ am Institut für Biologie der Humboldt-Universität zu Berlin unter der Leitung von Prof. Dr. Annette Upmeier zu Belzen erstellt. Ich möchte mich ganz herzlich bei Frau Prof. Upmeier zu Belzen bedanken, da sie mir die Möglichkeit gab die Abschlussarbeit in einem so spannenden Projekt zu erstellen. Ich möchte mich auch bei Herrn Prof. Priemer für seine Bereitschaft zur Zweitbegutachtung bedanken.

Ganz herzlich möchte ich mich bei Lucas bedanken, es war ein großes Vergnügen mit ihm an diesem Projekt zu arbeiten. Er half mir sehr dabei die Struktur für den didaktischen Teil dieser Arbeit zu entwickeln.

Ich möchte mich auch ganz herzlich bei meiner Familie bedanken. Sie haben immer an mich geglaubt und mich das auch spüren lassen. Vielen Dank Opa, Mutti und Jule!

Ganz besonders aber möchte ich mich bei dir, Jennifer bedanken. Ohne dich wäre diese Arbeit so nicht möglich gewesen!

Danke auch an Sina und Niko, *Apis mellifera*, *Drosera binata*, *Nepenthes spp.*, *Dionaea muscipula*, *Drosera adelae*, *Drosera aliciae*, *Drosera capensis*, im Geiste auch *Drosera rotundifolia* und allen, die ich hier noch vergessen habe!

Anhang

Quellcode des DataLabs

HU-Box:

<https://box.hu-berlin.de/d/e981200a00b24e5483ec/>

Passwort: MA_Beuster

Der Quellcode in der HU-Box entspricht dem Stand der Arbeit. Der Code in den anderen Quellen wird sich durch die weitergehende Entwicklung mit der Zeit verändern.

Github:

<https://github.com/WeirdCircumstances/datalab.git>

Docker-Image:

<https://hub.docker.com/repository/docker/docr/datalab/general>

Zugangsdaten der bestehenden Installation für Admin Panel und CMS:

Login: explorer

Passwort: :^hD :?xag##^w; `=9pvrb

Bauen und Deployen

Variablen definieren

Die Datei `.env.template` dient als Vorlage, um die beiden Dateien `.env.dev` und `.env.prod` zu erstellen. Die Felder mit `***` müssen mit den passenden Einträgen versehen werden.

Entwicklungsumgebung starten:

`docker compose up --build`

Um das Produktivsystem zu bauen gibt es zwei Möglichkeiten:

Möglichkeit 1:

```
docker compose build -f compose-production.yml datalab
docker push <dockerhub_username>/datalab
```

Möglichkeit 2:

Die Produktivumgebung kann auch auf dem Zielserver gebaut werden. Dazu müssen alle Projektdateien zuvor auf den Server übertragen werden. Das Script `./upload.sh` hilft bei diesem Prozess.

```
docker compose build -f compose-production.yml datalab
```

Produktivumgebung ausführen

```
docker compose -f compose-production.yml up
```

Data Literacy Competencies Matrix (Ridsdale et al., 2015)

Appendix 1 - Data Literacy Competencies Matrix

This matrix consists of key data literacy ability/knowledge areas, and the corresponding competencies and tasks required for each.

Definition: Data literacy is the ability to collect, manage, evaluate, and apply data; in a critical manner.

Key Ability/Knowledge Area	Competency	Knowledge/Tasks		LEGEND:	Core Competencies	Advanced Competencies
		Conceptual Framework	Introduction to Data			
Data Collection	Data Discovery and Collection Evaluating and Ensuring Quality of Data and Sources	Knowledge and understanding of data applications of data	Knowledge and understanding of the uses and applications of data			
Data Management	Data Organization Data Manipulation Data Conversion (from format to format) Metadata Creation and Use Data Curation - Security, and Re-Use Data Preservation	Performs data exploration Critically assesses sources of data for inauthenticity	Identifies useful data Critically evaluates quality of datasets for errors or problems		Collects data	
					Organizes data Cleans data	
					Identifies outliers and anomalies Converts data from one format or file type to another	
					Assesses appropriate metadata descriptions to original data sets	
					Assesses data security requirements (e.g. restricted access, protected drives, etc.)	
					Curates data Preserves data	
					Assesses methods and tools for data preservation	
Data Tools	Knowledge of data analysis tools and techniques	Selects appropriate data analysis tool or technique	Applies data analysis tools and techniques			
Basic Data Analysis	Develops analysis plans	Applies analysis methods and tools	Conducts exploratory analysis			
Data Interpretation (Understanding Data)	Reacts to understandings of new information	Identifies key take-away points, and integrates them with other data	Identifies user gaps within the data			
Data Evaluation	Identifying Problems Using Data	Uses data to identify problems in practical situations (e.g. workplace efficiency)	Uses data to identify higher level problems (e.g. marketing, economics, etc.)			
	Data Visualization	Creates meaningful tables to organize and visually present data	Creates meaningful graphical representations of data			
	Presenting Data (Verbally)	Assesses the desired outcome(s) for presenting the data	Assesses audience needs and familiarity with subjects			
	Data Driven Decisions Making (DDDM) (Making decisions based on data)	Prioritizes information gained from data	Converts data into actionable information			
					Evaluates the merit and impacts of possible solutions/decisions	
					Implements decisions/solutions	
Critical Thinking	Aware of high level issues and challenges associated with data					
Data Culture	Recognizes the importance of data					
Data Application	Data Ethics Data Citation Data Sharing	Aware of legal and ethical issues associated with data Knowledge of widely-accepted data citation methods Assesses methods and platforms for sharing data Evaluating Decisions Based on Data	Supports an environment that fosters critical use of data for learning, research, and decision-making Applies and works with data in an ethical manner Creates correct citations for secondary data sets Shares data legally and ethically Conducts a analysis of follow-up data based upon data			
					Compares results of analysis with other findings Presents arguments and/or outcomes Utilizes meaningful tables and visualizations to communicate data clearly and coherently	
					Raises original conjecture or decisions or implements new decisions/outputs	

Eigenständigkeitserklärung

Lebenswissenschaftliche Fakultät

Institut für Biologie

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig verfasst habe und sämtliche Quellen, einschließlich Internetquellen, die unverändert oder abgewandelt wiedergegeben werden, insbesondere Quellen für Texte, Grafiken, Tabellen und Bilder, als solche kenntlich gemacht habe.

Ich versichere, dass ich die vorliegende Abschlussarbeit noch nicht für andere Prüfungen eingereicht habe.

Mir ist bekannt, dass bei Verstößen gegen diese Grundsätze ein Verfahren wegen Täuschungsversuchs bzw. Täuschung gemäß der fachspezifischen Prüfungsordnung und/oder der Fächerübergreifenden Satzung zur Regelung von Zulassung, Studium und Prüfung der Humboldt-Universität zu Berlin (ZSP-HU) eingeleitet wird.

Berlin,

Benjamin Beuster