HAW HAMBURG

MASTER THESIS
Jonathan Ströbele

Der Data Hub: ein Geoinformationssystem für reproduzierbare Datenverarbeitung, informiert durch epidemiologische Bedarfe

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Engineering and Computer Science
Department Computer Science

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
Hamburg University of Applied Sciences

Jonathan Ströbele

Der Data Hub: ein Geoinformationssystem für reproduzierbare Datenverarbeitung, informiert durch epidemiologische Bedarfe

Masterarbeit eingereicht im Rahmen der Masterprüfung
im Studiengang *Master of Science Informatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Thomas Clemen
Zweitgutachter: Prof. Dr. Marina Tropmann-Frick

Eingereicht am: 23.11.2023

Jonathan Ströbele

Thema der Arbeit

Der Data Hub: ein Geoinformationssystem für reproduzierbare Datenverarbeitung, informiert durch epidemiologische Bedarfe

Stichworte

GIS, Reproduzierbar, Risikobewertung, EWARS, Epidemiologie, Datenmanagement

Kurzzusammenfassung

Die Gefahren durch tropische Infektionskrankheiten nehmen weltweit zu. Die Gesundheitssysteme von ressourcenlimitierten Ländern des globalen Südens sind davon besonders betroffen, bedingt durch den Klimawandel gilt dies auch zunehmend für die Länder des globalen Nordens. Innovative, datengetriebene Systeme leisten einen wichtigen Beitrag zur Früherkennung von epidemischen Bedrohungslagen und in der Planung von Interventionsmaßnahmen. Durch die Verbindung und Aufbereitung verschiedener Daten aus unterschiedlichsten Quellen können, in Hinblick auf den One-Health-Gedanken, Entscheidungsträgern hilfreiche Kontextinformationen zur Bewertung von Ausbruchsgeschehen bereitgestellt werden. In dieser Arbeit wurde ein Geoinformationssystem (Data Hub) entwickelt, das unterschiedlichste Datenarten und -quellen integriert und anhand reproduzierbarer Risikobewertungsprozesse ein Entscheidungsunterstützungssystem für Epidemiologen darstellt. Durch die interdisziplinäre Zusammenarbeit von Epidemiologie und Informatik konnte auf dieser Basis am Beispiel des Denguefiebers in Tansania ein Frühwarnsystem entwickelt werden. Darüber hinaus bietet der generische Aufbau des Data Hubs Potenziale, spatio-temporale Daten für weitere Anwendungsszenarien in anderen Domänen einzusetzen.

Jonathan Ströbele

Title of Thesis

The Data Hub: a geographic information system for reproducible data analysis, informed by epidemiological needs

Keywords

GIS, Reproducible, Risk assessment, EWARS, Epidemiology, Data management

Abstract

The health burden of tropical infectious diseases is increasing worldwide. Healthcare systems of resource-limited countries in the Global South are predominantly affected, a trend that is now extending to the Global North due to climate change. Innovative, data-driven systems can contribute to the early detection of disease outbreaks and public health threaths, and inform the planning of countermeasures. By linking and processing different data inputs from various sources, decision-makers can be provided with valuable contextual information for assessing outbreak risks in line with the One Health concept. Through this work, a geoinformation system (Data Hub) was developed that integrates a variety of data types and sources and provides a decision support system for epidemiologists based on reproducible risk assessments frameworks. On this basis, the interdisciplinary collaboration between epidemiology and computer science enabled the design of an early warning system using the example of dengue fever in Tanzania. In addition, the generic structure of the Data Hub holds potential for further application scenarios in other domains with respect to spatio-temporal data.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listingsverzeichnis

Abkürzungen

AoI Area of Interest

API Application Programming Interface

BNITM Bernhard-Nocht-Institut für Tropenmedizin

CLI Command-line-interface

DHS Demographic and Health Surveys

EIOS Epidemic Intelligence from Open Sources

EOSC European Open Science Cloud

ESIDA Epidemiological Surveillance for Infectious Diseases in sub-Saharan Africa

ETL Extract-Transform-Load

EWARS Early Warning, Alert and Response System

FSF Free Software Foundation

GEE Google Earth Engine

GIS Geoinformationssystem

HAW Hamburg Hochschule für Angewandte Wissenschaften Hamburg

MVC Model-View-Controller

OGC Open Geospatial Consortium

OKF Open Knowledge Foundation

OOP Objektorientierte Programmierung

ORM Object–Relational-Mapping

OSD Open Source Definition

OSI Open Source Initiative

OSM OpenStreetMap

PoI Point of Interest

RRA Rapid Risk Assessment

WHO World Health Organization

1 Einleitung

Infektionskrankheiten werden zunehmend zu einem größeren Problem für Gesundheitssysteme [24]. Ein prominentes Beispiel hierfür ist die COVID-19 Pandemie, allerdings gibt es auch Ausbrüche anderer Krankheiten, wie Ebola, Zika-Virus oder Denguefieber [8]. Zunehmende anthropogene Einflüsse auf das weltweite Ökosystem, wie intensive Landwirtschaft, Abholzung und Urbanisierung können zur weiteren Häufung von Zoonosen führen, bedingt durch den sich intensivierenden Kontakt von Mensch und Tier [73]. Zusätzlich kann die Klimaveränderung das Risiko von Krankheitsausbrüchen erhöhen und zu Verschiebungen von endemischen Krankheitsgebieten führen [63], was Gesundheitssysteme weltweit vor neue Herausforderungen stellt.

Zwar gab es in den Ländern des globalen Südens in den letzten Jahren Verbesserungen in der Gesundheitsversorgung, sie sind aber im Vergleich zu Industrienationen dennoch stark von Krankheiten und unzureichender Gesundheitsversorgung betroffen [77]. In Subsahara-Afrika stellen tropische Infektionskrankheiten für einen großen Teil der Bevölkerung ein besonders hohes Risiko dar [37]. Durch geringe Testkapazitäten und mangelnde Laborinfrastrukturen wird die frühzeitige und zuverlässige Detektion von Ausbruchsgeschehen erschwert [49]. Aber auch für den globalen Norden sind tropische Infektionskrankheiten zunehmend problematisch [36], wie erste lokal erworbene Fälle des Denguefiebers in Südfrankreich in 2022 [21] oder ein lokaler Ausbruch von Malaria in Florida im Jahr 2023 [15] zeigen.

Angesichts dieser wachsenden Bedrohungslagen sind innovative und datengetriebene Warn- und Informationssysteme ein essenzieller Baustein für das Gesundheitswesen [58]. Solche Systeme erleichtern die frühzeitige Erkennung und Planung von Interventionen. Durch die Kombination und Integration verschiedener Datenquellen unterliegen sie allerdings einer hohen Komplexität. Dennoch bietet die Digitalisierung große Chancen für die Stärkung von Gesundheitssystemen, wie der vielseitige Einsatz während der Corona-Pandemie gezeigt hat [13]. Sie ist ein Bestandteil der One-Health-Initiative, in der Mensch, Tier und Umwelt als gesamtheitliches System betrachtet werden und

entsprechende interdisziplinäre Ansätze für eine nachhaltige Entwicklung des globalen Gesundheitssektors zwingend erforderlich sind [52].

In dem interdisziplinären Forschungsprojekt Epidemiological Surveillance for Infectious Diseases in sub-Saharan Africa (ESIDA) wird die Frage untersucht, inwieweit datengesteuerte Informationssysteme Gesundheitsakteure bei der Früherkennung von Krankheitsausbrüchen unterstützen können. Untersucht wird exemplarisch das Denguefieber innerhalb von Tansania. Das Projekt vernetzt dabei internationale Partner aus „Epidemiologie, klinischer Forschung, Biosicherheit, Public Health, Umweltwissenschaften sowie Computer und Data Science“[1]. Die vorliegende Masterarbeit behandelt in diesem Kontext die technische Entwicklung einer Datenarchitektur und eines Informationssystem für diese Bedarfe (Arbeitspaket 4, ESIDA).

1.1 Denguefieber

Das Denguefieber ist ein hämorrhagisches Fieber, das durch das Dengue-Virus ausgelöst wird und vor allem durch die Gelbfiebermücke (*Aedes aegypti*, Abb. 1.1) verbreitet wird [75]. Die weiblichen Mücken nehmen dabei das Virus von infizierten Menschen bei der Blutmahlzeit auf und geben es bei folgenden Stichen weiter. Eine direkte Infektion von Mensch zu Mensch findet nicht statt. Die Mücke fungiert also als Krankheitsüberträger, bzw. als sogenannter *Vektor*. Die meisten Infektionen verlaufen asymptomatisch und milde Symptome klingen bereits nach wenigen Tagen wieder ab [75]. In 2 % bis 4 % der Fälle kann es allerdings zum schweren Verlauf kommen, mit hohem Fieber, Ausschlag, Kopf- und Gliederschmerzen, die den alternativen Namen *Knochenbrecherfieber* unterstreichen [75]. Bei diesen schweren Verläufen liegt die Sterblichkeit bei 1 % bis 5 % [75]. Im Jahr 2019 sind schätzungsweise 36 000 Menschen durch Dengue verstorben [84]. Modellierungen gehen davon aus, dass sich bis zu 390 Millionen Menschen im Jahr infizieren [6]. Eine gezielte Therapie bei einer

Abbildung 1.1: *Aedes aegypti* bei der Blutmahlzeit [39].

[1] https://www.haw-hamburg.de/forschung/forschungsprojekte-detail/project/project/show/esida/

Infektion existiert nicht, lediglich die symptomatische Behandlung in einem Krankenhaus mit Flüssigkeitsgabe und Schmerzlinderung [75].

Als tropische Krankheit tritt das Denguefieber hauptsächlich in tropisch und subtropischen Gebieten auf [44], wie in Abb. 1.2 zu sehen. Heutzutage leben bereits über 50 % der Weltbevölkerung in potenziellen Dengue-Gebieten [44, 70]. Messina et al. gehen davon aus, dass bis 2080 die Bevölkerung in Risikogebieten weiter zunehmen wird [44]. Dies spiegelt sich auch wider in der zunehmenden Häufung von Dengue-Ausbrüchen, sowie der Steigerung der Dengue-Inzidenz um das 30-fache in den letzten 50 Jahren [54].

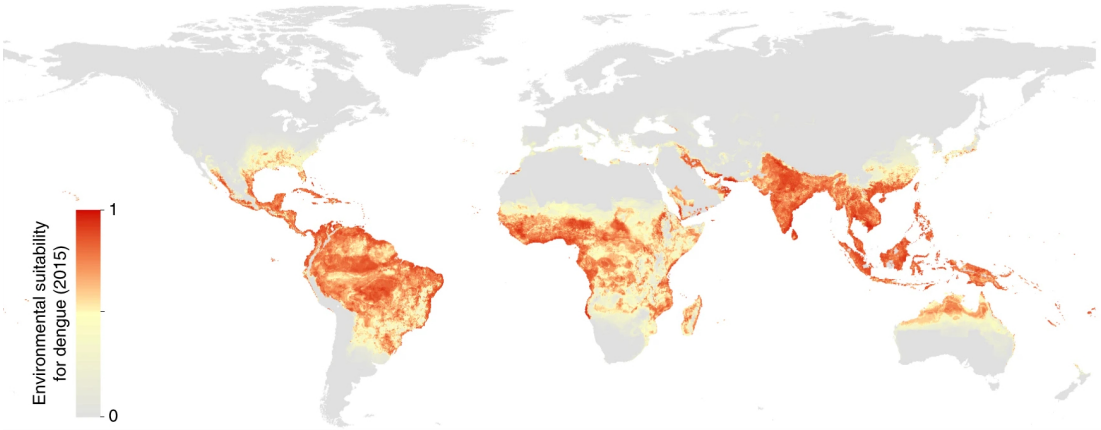


Abbildung 1.2: Modellierte Umweltverträglichkeit von Dengue [44].

Impfstoffe gegen das Dengue-Virus sind noch in der Erprobung, bzw. Zulassung, und nicht großflächig verfügbar [55]. Daher besteht aktuell lediglich die Möglichkeit des passiven Schutzes durch das Individuum selbst, bspw. durch das Tragen langer Kleidung und die Benutzung von Insektensprays, um Stiche zu verhindern. Zusätzlich gibt es die aktive Vektor-Kontrolle, bei der Brutstätten der Mücke vernichtet werden oder Insektizide großflächig versprüht werden [1]. Zukünftig wird erwartet, dass sich das Denguefieber zunehmend weiter ausbreiten und damit eine hohe Belastung für die Gesundheitssysteme weltweit darstellt.

Die relevanten epidemiologischen Parameter für das Auftreten von Denguefieber ergeben sich vor allem aus Umweltfaktoren, die für die Überlebensfähigkeit der Mücke relevant sind. Hierunter fallen etwa Temperatur, Luftfeuchtigkeit oder Niederschlag [2]. Zusätzlich spielen aber auch sozioökologische Faktoren eine Rolle, wie Bevölkerungsdichte, Bildungsstand oder Landnutzung [2].

1.2 Fragestellung

Der konzeptionelle Überblick des ESIDA-Projektes ist in Abb. 1.3 dargestellt. Es wird ein System entwickelt, in dem verschiedene, relevante Datenquellen integriert werden. Dabei werden sie räumlich auf tansanische Verwaltungseinheiten zugeschnitten und zeitlich entsprechend ihrer Auflösung eingebunden. Auf Basis dieser Daten soll ein Risiko-Assessment durchgeführt werden, mit dessen Ergebnis Entscheidungsträger informiert werden können. Zusätzlich soll das System für Epidemiologen beim Aufkommen von Krankheitsmeldungen ein Informationsportal bereitstellen, mit dem der Ausbruch anhand der integrierten Daten eingeordnet und bewertet werden kann.

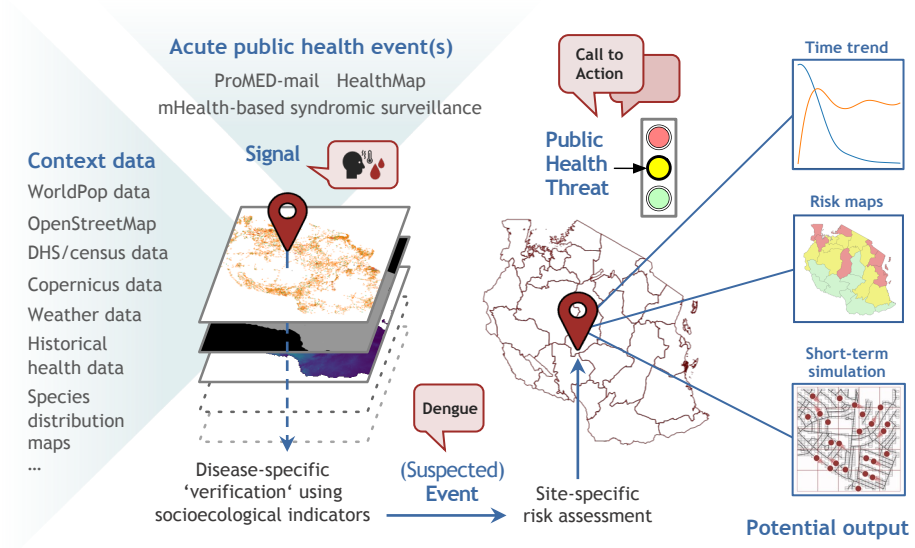


Abbildung 1.3: Konzept des Data Hubs im Rahmen des ESIDA-Projektes im Überblick [11].

Im Rahmen dieser Masterarbeit steht dabei die technische Umsetzung des Data Hubs im Vordergrund und *nicht* die epidemiologische Validierung der Ergebnisse. Konkret handelt es sich dabei um die Frage der Realisierbarkeit eines solchen Systems. Im Fokus stehen die folgenden Fragestellungen:

1. Wie können verschiedene heterogene Daten in ein solches System eingebunden und homogen verarbeitet werden?

2. Kann das System von Epidemiologen zur Durchführung von reproduzierbaren Risikobeurteilungs-Algorithmen auf Grundlage der aggregierten Daten verwendet werden?
3. Lässt sich das System generisch entwickeln, sodass es für weitere Anwendungsszenarien verwendet werden kann?

Insbesondere der dritte Punkt ist von Interesse. Das System wird zwar anhand der Bedarfe des Projektes im Rahmen von Denguefieber in Tansania informiert, jedoch ergibt sich die Notwendigkeit von einfach zu installierenden Werkzeugen zur reproduzierbaren Verarbeitung geografischer Daten nicht nur in der Epidemiologie, sondern auch in vielen anderen Disziplinen, wie der Hydrologie [41] oder der Bioinformatik [67].

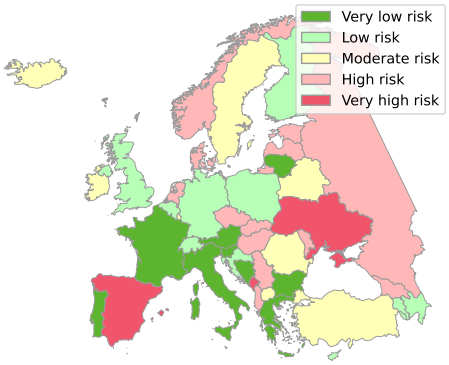
2 Grundlagen

In den folgenden Abschnitten dieses Kapitels werden zuerst die epidemiologischen und anschließend die informatischen Hintergründe vorgestellt. Damit soll eine fundierte Grundlage zur Einordnung des Data Hubs geschaffen werden, auf Basis des aktuellen Stands der Forschung.

2.1 Risiko

Unter dem Begriff Risiko versteht man gemeinhin die Kombination aus der Eintrittswahrscheinlichkeit eines Ereignisses und der Schwere des zu erwartenden Schadens. Die Eintrittswahrscheinlichkeit ist dabei die *Probability* und die Schwere der *Impact* [23]. Im Kontext der Epidemiologie beschreibt ein Risiko bspw. die Gefahr, dass bestimmte Bevölkerungsgruppen sich mit einem Erreger infizieren.

Treten innerhalb kurzer Zeit vermehrt identische Krankheitsfälle unbekannter Herkunft auf, müssen das weitere Vorgehen und Gegenmaßnahmen schnell bestimmt werden. Dafür muss das Risiko eines Infektionsausbruchs eingeschätzt und entsprechend verständlich an Entscheidungsträger kommuniziert werden. Dieser Vorgang wird als Rapid Risk Assessment (RRA) bezeichnet. Hierfür müssen schnellstmöglich alle verfügbaren Informationen, sowie weitere relevante Kontextdaten zusammengestellt und ausgewertet werden [23, 83]. Für die Kommunikation eines Risikos bieten sich verschiedene Möglichkeiten an. Darunter fallen Risiko-Karten, wie in Abb. 2.1a dargestellt, oder eine detailliertere Bewertung mittels einer Risiko-Matrix, wie in Abb. 2.1b gezeigt.



(a) Eindimensionale Darstellung eines hypothetischen Risikos in Europa als Choroplethenkarte.

Propability

Impact	Very low	Low	Moderate	High
Very low	Very low risk	Low risk	Low risk	Moderate risk
Low	Low risk	Low risk	Moderate risk	Moderate risk
Moderate	Low risk	Moderate risk	Moderate risk	High risk
High	Moderate risk	Moderate risk	High risk	Very high risk

(b) Zweidimensionale Risiko-Matrix, die Schwere und Eintrittswahrscheinlichkeit ablesbar macht, abgewandelt von [23].

Abbildung 2.1: Risikokommunikation mittels Grafiken.

Elementar an einem RRA ist, dass die Einschätzungen innerhalb kurzer Zeit zur Verfügung stehen. Zusätzlich müssen Unsicherheiten in der Datengrundlage kenntlich gemacht werden. Der Prozess muss reproduzierbar und transparent sein, sodass ein hohes Maß an Vertrauen in die Ergebnisse ermöglicht wird.

2.2 Surveillance

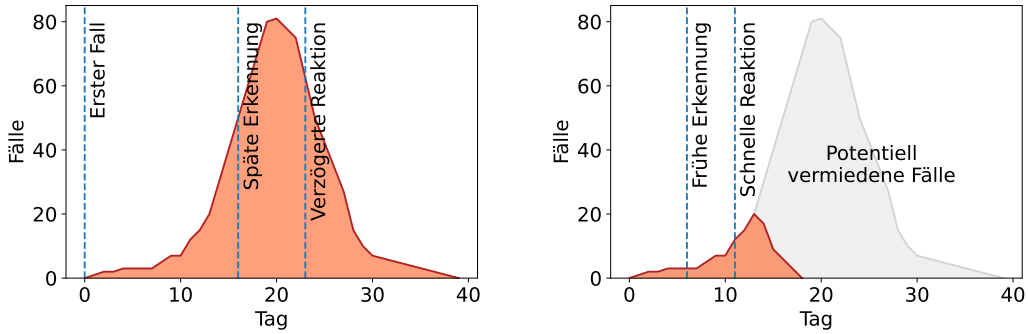
Die epidemiologische Überwachung, bzw. Surveillance, ist die „ongoing systematic collection, analysis, and interpretation of data.“ [38] Durch die kontinuierliche Datenauswertung kann die Gesundheitslage fortlaufend überwacht werden oder neu auftretende Probleme erkannt werden und somit Entscheidungsträger informiert und Interventionen geplant werden. Man unterscheidet dabei die aktive und passive Surveillance. Im Fall der aktiven Surveillance werden gezielt Gesundheitsmetriken abgefragt und aufbereitet, wohingegen bei der Passiven Gesundheitsdaten automatisiert übertragen und ausgewertet werden.

Für eine schnelle Reaktion und Auswertung von Daten ist eine digitale Lösung für die Surveillance wichtig [69]. Ein Beispiel für eine solche Lösung ist das vom Helmholtz-Zentrum

für Infektionsforschung entwickelte SORMAS[1], in dem Falldaten durch Gesundheitsämter erhoben, zusammengeführt und ausgewertet werden können. Die als Open Source bereitgestellte Software befindet sich bereits in mehreren Ländern im Einsatz.

Zentraler Bestandteil von Surveillance-Systemen ist ein Early Warning, Alert and Response System (EWARS) [81]. Ein solches System soll im Fall eines Ausbruchs verschiedene Daten schnell zusammenführen, auswerten und damit Entscheidungsträger umgehend informieren. Dadurch können Interventionen besser geplant und zielgerichtet durchgeführt werden. Mit „EWARS in a box“[2] bietet die World Health Organization (WHO) ein Interventions-Kit an, mit dem in einem akuten Krisenfall schnell mobile Geräte zur Datenerhebung, sowie Software zur Analyse im Feld eingesetzt werden können [82].

Eine zeitnahe und direkte Implementierung von Gegenmaßnahmen als Reaktion auf ein Ausbruchsgeschehen ist von höchster Priorität. Wie in Abb. 2.2a gezeigt, führt eine späte Erkennung eines Ausbruchs, und die damit einhergehende verzögerte Reaktion zu einem deutlich größeren Ausbruch. Ist dagegen der Ausbruch bereits früh mittels eines EWARS erkannt und sind direkt entsprechende Gegenmaßnahmen eingeleitet worden, lassen sich der Ausbruch eindämmen und damit viele Infektionen vermeiden, wie in Abb. 2.2b zu sehen.



(a) Verspätete Reaktion durch verzögerte Erkennung des Ausbruchs.

(b) Beschleunigte Reaktion auf einen Ausbruch durch frühere Erkennung.

Abbildung 2.2: Effekt eines EWARS, Abbildung adaptiert von [53].

Hervorzuheben ist, dass ein EWARS nicht ausschließlich auf den erhobenen Gesundheitsmeldedaten basieren sollte. Krankheitsausbrüche hängen oft mit weiteren Indikatoren eng

[1] https://sormas.org/
[2] https://www.who.int/emergencies/surveillance/early-warning-alert-and-response-system-ewars

zusammen, wie bspw. Klima- und Wetterdaten [4]. Daher ist es zwingend, dass innovative Systeme zusätzliche Datenquellen in Bewertungsprozesse einbeziehen.

2.2.1 Event-based Surveillance

Neben den offiziellen Meldezahlen können inoffizielle Berichte, wie bspw. aus Medien oder Social-Media-Meldungen, Hinweise auf ein sich entwickelndes Ausbruchsgeschehen liefern [16]. Diese Meldungen sind oftmals schneller verfügbar, als die der langsamer agierenden offiziellen Institutionen. In dem seit 1994 existierendem Projekt ProMED[3] werden solche Meldungen täglich zusammengestellt. Es stellt damit eine wichtige Informationsquelle für Gesundheitsforschende dar. Ein weiteres Projekt ist HealthMap[4], in dem stündlich eine Vielzahl verschiedener Online-Quellen automatisiert gesammelt werden und die Ergebnisse auf einer interaktiven Karte verortet werden.

Beide Projekte sind Bausteine der Event-based Surveillance, die von der WHO in ihrer Epidemic Intelligence from Open Sources (EIOS) Initiative gebündelt werden. Die Initiative hat das Ziel, Krankheitsausbrüche frühzeitig zu erkennen und entsprechend schneller Interventionen einleiten zu können. Meldungen aus diesen Systemen können Eingaben in ein EWARS darstellen, in welchem sie hinsichtlich ihrer Plausibilität und Gefahr bewertet werden können.

2.3 Open Source

Im Jahr 1985 gründete Richard Stallmann die Free Software Foundation (FSF) und ein Jahr später veröffentlichte er in diesem Rahmen die *Free Software Definition*. Wobei *free* in diesem Kontext nicht für eine kostenlose Weitergabe von Software steht, sondern für die Freiheit, die Software einsehen, modifizieren und weitergeben zu können [62]. In der heutigen Version umfasst die Free Software Definition vier Regeln[5]:

1. Die Freiheit, die Software ausführen zu können.
2. Die Freiheit, die Software verstehen zu können, was eine Offenlegung des Quellcodes impliziert.
3. Die Freiheit, die Software weitergeben zu können.

[3] https://promedmail.org/
[4] https://www.healthmap.org/
[5] https://www.gnu.org/philosophy/free-sw.html.en#four-freedoms

4. Die Freiheit, modifizierte Versionen der Software weitergeben zu können.

Zum einen gibt es die sprachliche Doppeldeutigkeit des englischen Wortes *free* (gratis, Freiheit), zum anderen hatten Firmen, die ein wirtschaftliches Interesse an der Vermarktung ihrer Software hatten, Schwierigkeiten bei der Anwendung der Free Software Definition [40]. Es gab durchaus auch *freie* Software, die aber keinen frei zugänglichen Quellcode bereitstellte.

Unter dem Begriff *Open Source* versteht man gemeinhin Software, deren Quellcode frei zugänglich ist. Im Jahr 1998 wurde von den beiden Pionieren der Open-Source-Bewegung Bruce Perens und Eric S. Raymond die Open Source Initiative (OSI) gegründet. Auf Basis der *Debian Free Software Guidelines*[6] veröffentlichte Perens die Open Source Definition (OSD)[7], welche über die OSI verwaltet wird.

Die OSD enthält 10 Kriterien, die für eine Open-Source-Software relevant sind. Die zentralen Aspekte sind dabei die freie Weitergabe ohne Lizenzgebühren (1), dass der Quellcode jedem Nutzer zur Verfügung steht (2) und die uneingeschränkte Weitergabe von Veränderungen an der Software (3). Diese Definition überschneidet sich damit im Wesentlichen mit der der FSF. Free Software nach der FSF kann als Untermenge von Open-Source-Software betrachtet werden, bspw. könnte eine Digital-Rights-Management Software zwar Open Source sein, würde aber die Freiheit des Anwenders hindern. Richard Stallmann beschreibt diese Differenzierung mit den Worten:

> Nearly all open source software is free software; the two terms describe almost the same category of software. But they stand for views based on fundamentally different values. Open source is a development methodology; free software is a social movement. For the free software movement, free software is an ethical imperative, because only free software respects the users' freedom. [71]

Die FSF und die OSI bewerten jeweils Lizenzen, die unter ihrer jeweiligen Definition als gültig angesehen werden. Dabei gibt es einzelne Unterschiede, aber viele Lizenzen werden von beiden Bewegungen anerkannt. Darunter fallen etwa die bekannten Apache-, GPL- oder MIT-Lizenzen[8]. Die MIT-Lizenz wird auf der Code-Hosting-Platform GitHub unter

[6] https://www.debian.org/social_contract#guidelines
[7] https://opensource.org/osd/
[8] Gegenüberstellung verschiedener Lizenzen und der jeweiligen Anerkennung durch OSI und FSF: https://spdx.org/licenses/

den Projekten mit explizit angegebener Lizenz mit Abstand am meisten verwendet mit 44,69 %, gefolgt von GPL (v2 oder v3) mit 20,07 % und Apache mit 11,19 % [5].

Die Wahl einer offenen Lizenz ist gerade für den Einsatz von kritischer Infrastruktur im Krisenmanagement von großer Bedeutung. In Notsituationen dürfen Lizenz- oder Nutzungsrechte nicht den Einsatz der Software behindern. Dies könnte bspw. durch geografisch limitierte Nutzungsrechte oder Exportbeschränkungen in das betroffene Land geschehen. Zusätzlich können hohe Lizenzkosten den Einsatz solcher Software in Regionen mit begrenzten Ressourcen unmöglich machen [19]. Manche Softwarehersteller bieten zwar vereinfachte Lizenzmodelle für Krisensituationen an, wie bspw. Esri mit ihrem „Desaster Response Program“[9]. Allerdings besteht hier nach wie vor ein organisatorischer Aufwand der im Notfall wertvolle Ressourcen bindet, die dadurch nicht dem Management des eigentlichen Notfalls zur Verfügung stehen. Daher ist in diesem Kontext eine offene Lizenz elementar, die dem Anwender garantiert, dass es im Anwendungsfall zu keinen Hürden kommt, die dem Einsatz der Software und damit der Linderung der Krise im Wege stehen.

2.4 Open Data

Für die wissenschaftliche Arbeit, sowie zeitnahe Interventionen in Krisen [32], sind Daten essenziell. Sie beschreiben und fixieren gesammelte Beobachtungen, Erkenntnisse und Ergebnisse. Mittels dieser aufgezeichneten und bereitgestellten Daten findet der Austausch zwischen Forschenden statt und ermöglicht so den wissenschaftlichen Fortschritt [25]. Dieser Prozess gilt nicht nur zwischen korrespondierenden Wissenschaftlern, sondern auch für die Weitergabe von Informationen an spätere Generationen.

Allerdings sind diese Daten nicht immer frei verfügbar. Sie können nur kleinen Nutzerkreisen bereitgestellt oder durch restriktive Lizenzmodelle nicht verwendbar sein. Zusätzlich können Daten auch in nicht maschinenlesbaren Formaten existieren, die eine Weiterverwendung erschweren bis unmöglich machen. Oftmals gibt es nicht genügend Anreize für Wissenschaftler, ihre Daten dokumentiert und organisiert der Öffentlichkeit zur Verfügung zu stellen. Fehlende Anerkennung, die Sorge, das Potenzial für weitere Publikationen zu verschenken oder mangelnde Zeit können Gründe hierfür sein [45, 56].

[9] https://www.esri.com/en-us/disaster-response/overview

Die Open Knowledge Foundation (OKF) wurde 2004 aus einer wachsenden Gemeinschaft heraus gegründet, die sich der *Open Science* verpflichtet sieht. Von ihren Mitgliedern wird seit 2005 die *Open Definition*[10] gepflegt, in der die Anforderungen an einen offenen Umgang mit Daten, bzw. *Open Data*, definiert wird. Das Dokument nimmt wiederum Bezug auf die zuvor vorgestellte Free Software Definition und die OSD. Zentraler Inhalt ist, dass die Daten zum einen frei, maschinenlesbar und in einem offenen Format über das Internet bereitgestellt werden, und zum anderen die Daten unter einer offenen Lizenz stehen, die die Verarbeitung und Weitergabe zu jedem Zweck erlaubt. Lediglich eine Attribuierung und die Weitergabe unter der gleichen oder einer ähnlichen Lizenz darf eingefordert werden. Die OKF selbst fasst dies so zusammen: „Open means anyone can freely access, use, modify, and share for any purpose (subject, at most, to requirements that preserve provenance and openness).“

Die Open Definition reiht sich neben einigen anderen Initiativen, die sich zu Open Data, bzw. Open Science, bekennen ein. Beispiele umfassen die Budapest Open Access Initiative[11] (2001), die Berliner Erklärung[12] (2003) oder die Panton Principles[13] (2010). Die Praxis von Open Data wird sowohl von Seite der Wissenschaftler, als auch der Geldgeber und Fachjournals, immer häufiger eingefordert [45]. Die so neu gewonnenen Möglichkeiten der Validierung von Ergebnissen und Verlinkung verschiedener Daten stärken den wissenschaftlichen Betrieb [56, 78].

Im Zuge dieser Entwicklung haben sich Datenjournale etabliert [14], in denen keine vollständige wissenschaftliche Paper veröffentlicht werden, sondern Datensätze, unter Berücksichtigung eines Peer-Review. Dies ermöglicht es Wissenschaftlern für ihre gesammelten und aufbereiteten Daten attribuiert zu werden und lindert damit einige der zuvor erwähnten Probleme. Zusätzlich gibt es Datenrepositorien, wie Zenodo[14], auf denen Wissenschaftler ihre Daten oder ihren Quellcode hinterlegen können und darüber zitiert werden können.

2.4.1 FAIR

Die zuvor genannten Entwicklungen führen dazu, dass große Datenmengen immer einfacher geteilt und zur Verfügung gestellt werden können. Dies bedeutet, dass auf Konsu-

[10] https://opendefinition.org/od/2.1/
[11] https://www.budapestopenaccessinitiative.org/read/
[12] https://openaccess.mpg.de/Berliner-Erklaerung
[13] https://web.archive.org/web/20220331215657/http://pantonprinciples.org/
[14] https://zenodo.org/

mentenseite nun ein höherer Druck besteht, passende und gute Daten zu identifizieren. Dafür sind gepflegte Metadaten zu den bereitgestellten Daten elementar. Aus diesem Hintergrund heraus hat sich 2014 eine interdisziplinäre Arbeitsgruppe gebildet, die die FAIR-Richtlinien entwickelt hat, die technische Regeln definieren, wie man Daten aufbereiten sollte, um sie effektiv zu teilen [79].

Findable Die Daten sind mit umfangreichen Metadaten versehen. Diese sollen sowohl für Menschen als auch Maschinen lesbar sein. Durch die maschinelle Lesbarkeit soll gewährleistet werden, dass die Daten von Suchmaschinen indexiert und gefunden werden können.

Accessable Die Daten sind über eine eindeutige Kennung abrufbar. Der Zugriff auf die Daten erfolgt über ein freies Protokoll.

Interoperable Die Beschreibung der Daten erfolgt in einer zugänglichen und etablierten Sprache. Zusätzlich können die Metadaten Verweise auf weitere relevante Daten enthalten.

Reusable Die Wiederverwendbarkeit der Daten soll durch umfangreiche, strukturierte Metadaten gewährleistet werden und die Daten stehen unter einer offenen Lizenz. Zusätzlich sollten domänenspezifische Kriterien berücksichtigt werden.

Die FAIR-Richtlinien sind weitläufig akzeptiert und befürwortet worden, sowohl von staatlicher Seite [27], als auch von der wissenschaftlichen Gemeinschaft [68]. Sie werden als integraler Bestandteil von Datenkatalogen angesehen [22].

2.5 Datenqualität und -nutzung

Bedingt durch die voranschreitende Digitalisierung wächst die Menge verfügbarer Daten immer weiter an [43]. Eng damit verbunden gibt es immer mehr Möglichkeiten, Daten zu veröffentlichen [14]. Dadurch wird es umso wichtiger, bei der Recherche von Datenquellen solche mit einer guten Qualität zu identifizieren. Die Qualität unterliegt hierbei den Bedarfen des jeweiligen Anwendungsszenarios und es muss daher individuell beurteilt werden, ob eine Datenquelle *gut genug* ist.

Die erhobenen, bzw. verwendeten Daten müssen folglich vorab eingehend untersucht werden, um festzustellen, ob sie für die geplante Verwendung geeignet sind. Dieser Prozess gehört zum Feld der *Data Science* und kann mit dem *Data Life Cycle* beschrieben werden [80]. Dieser umfasst die Schritte der Sammlung, Verarbeitung, Speicherung, Visualisierung und vor allem Interpretation der Daten. In der Domäne der Gesundheitswissenschaften unterliegt die systematische Auswertung und Aggregation von Daten ganz besonderen ethischen Leitlinien [66], da bei unsachgemäßer Verarbeitung sensitive Daten veröffentlicht werden könnten und somit Rückschlüsse auf einzelne Patienten möglich wären oder sogar durch die abgeleiteten Entscheidungen Menschenleben in Gefahr geraten könnten.

Besonders im Bereich der Open Data treten häufig Daten auf, die durch *Citizen Science* erhoben wurden [31]. In diesem Fall handelt es sich um Datenbestände, die durch Laien gesammelt und bereitgestellt wurden. Beispiele hierfür sind OpenStreetMap (OSM) für offene Kartenmaterialien oder iNaturalist, in dem weltweit die Artenvielfalt dokumentiert wird. Dies bedeutet, dass Daten unvollständig, veraltet oder schlichtweg falsch sein können. Im Falle von OSM zeigen verschiedene Studien, dass die Daten zwar eine hohe, aber dennoch nicht vollständige Abdeckung haben und die Qualität sich zwischen Regionen stark unterscheiden kann [7].

Für das wissenschaftliche Arbeiten ist es bei der Verwendung von Daten verschiedener Hintergründe umso wichtiger, dass die Verarbeitungsprozesse transparent und nachvollziehbar sind [3]. Dies erlaubt es die eigenen Ergebnisse bei Änderungen der Daten erneut ausführen zu können, sowie die Nachnutzung durch Dritte, und damit eine Stärkung der wissenschaftlichen Gemeinschaft.

2.6 Reproduzierbarkeit

Die Reproduzierbarkeit von Ergebnissen ist in der wissenschaftlichen Arbeit ein zentraler Aspekt. Dies gilt somit auch für Ergebnisse die mittels Programmcode berechnet wurden. Dies erfordert, dass der Programmcode zum einen zur Verfügung steht und zum anderen, dass er ausführbar ist. Dafür muss die Programmierumgebung für Dritte zur Verfügung stehen, sowie die Rohdaten auf denen der Code ausgeführt wurde. Trisovic et al. haben dazu über 2000 Forschungsprojekte im Harvard-Dataverse-Repository untersucht. In diesem Repository werden Daten und Quellcode zusammen als *replication packages* bereitgestellt, um eine Nachnutzung durch Dritte zu ermöglichen. Die Studie

hat ergeben, dass ohne Anpassungen nur 25 % der Projekte ausführbar waren. Außerdem enthielten 42,37 % der Projekte keine Form einer Dokumentation [76].

Dies zeigt, dass zusätzliche Richtlinien für Forschende benötigt werden, wie Quellcode zuverlässig weitergegeben werden kann, um eine Validierung und Nachnutzung zu ermöglichen. Von Sandve et al. wurden dazu 10 Regeln vorgeschlagen [67], die sich an Best Practices für Softwareentwicklung und wissenschaftlicher Arbeit orientieren. Die Regeln sind in Tabelle 2.1 aufgeführt und erläutert. Sie unterstreichen die Notwendigkeit, soweit möglich, alle Verarbeitungsabläufe in Form von Skripten und Programmcode durchzuführen. Manuelle Schritte sollen vermieden werden, bzw. mindestens dokumentiert werden. Die erstellten Codes und Daten sollten zudem öffentlich zugänglich sein. Um eine Ausführung zu erleichtern, sollte des Weiteren dokumentiert sein, welche Programmiersprachen oder Programme in welchen Versionen benötigt werden.

Tabelle 2.1: 10 Regeln für reproduzierbare Forschung [67].

#	Regel	Erklärung
1	For Every Result, Keep Track of How It Was Produced	Genau dokumentieren, welche Abläufe mit welchen Parameter ausgeführt wurden.
2	Avoid Manual Data Manipulation Steps	Alle Bearbeitungsprozesse sollten maschinell erfolgen.
3	Archive the Exact Versions of All External Programs Used	Updates von Software können zu einem anderen Verhalten führen, daher sollten die verwendeten Versionen dokumentiert werden.
4	Version Control All Custom Scripts	Erlaubt es, die Evolution des Codes nachzuvollziehen.
5	Record All Intermediate Results, When Possible in Standardized Formats	Ermöglicht einfacheres Debugging, um die Gründe von Diskrepanzen der Ergebnisse zu identifizieren.
6	For Analyses That Include Randomness, Note Underlying Random Seeds	Damit können zufallsbasierte Berechnungen reproduziert werden.

#	Regel	Erklärung
7	Always Store Raw Data behind Plots	Abbildungen können nachträglich geändert und neu erstellt werden.
8	Generate Hierarchical Analysis Output, Allowing Layers of Increasing Detail to Be Inspected	Bauen Ergebnisse auf Zwischenergebnissen auf, sollten diese auch erhalten bleiben.
9	Connect Textual Statements to Underlying Results	Interpretationen von Ergebnissen sollten eindeutig referenziert werden.
10	Provide Public Access to Scripts, Runs, and Results	Offener Zugang erlaubt Validierung.

2.7 Datenmanagement

Das Datenmanagement kann als Überbegriff des gesamten Lebenszyklus von Daten gesehen werden. Darunter fallen nicht nur die schon zuvor angesprochene Qualität und Verarbeitung, sondern auch die Akquise, langfristige Speicherung, die Sicherstellung von Zugriffsberechtigungen und die eventuelle Löschung von Daten [26].

Unter den Teilbereich der *Data Governance* fällt die Sicherung der Einhaltung von Regeln in den Verarbeitungsprozessen und Datenqualitätsstandards. Er wird vom *Datensteward*, bzw. dem Dateneigner, ausgeführt. Dies ist vor allem bei der Datenerhebung innerhalb einer Organisation der Fall.
Beim Zusammenführen verschiedener Datensätze aus unterschiedlichen Quellen ist es elementar, die Ursprünge der Datensätze zu kennen, sowie bei der Verarbeitung die durchgeführten Operationen zu dokumentieren. Dieser Teilbereich wird unter dem Begriff *Data-Lineage* verstanden.

2.7.1 Forschungsdatenmanagement

Das Forschungsdatenmanagement legt des Weiteren den Fokus auf die Organisation der, beim wissenschaftlichen Arbeiten anfallenden, Daten. Dies umfasst den gesamten Le-

benszyklus von Planung, Erhebung, Verarbeitung und Archivierung. Damit ist es ein wichtiger Bestandteil der guten wissenschaftlichen Arbeit [3, 61].

Speziell müssen dabei die unterschiedlichen Bedarfe verschiedener Disziplinen berücksichtigt werden. Bspw. hat die Speicherung archäologischer Daten andere Anforderungen [28] als die biologischer Daten [34]. Dennoch sollte eine gemeinsame Basis zwischen Disziplinen bestehen, damit Synergien genutzt werden können [61]. Darin begründet sich die Bestrebung, gemeinsame Dateninfrastrukturen zu schaffen. Innerhalb Deutschland geschieht dies in Form der Nationalen Forschungsdateninfrastruktur (NFDI) [61]. Für grundlegende Funktionalitäten soll eine gemeinsame Basis mittels Base4NFDI geschaffen werden, die Bausteine wie Authentifizierung und Identifizierung von Datensätzen erlauben soll. Aufbauend darauf ergeben sich weitere Bausteine für spezifische Disziplinen.

Auch auf europäischer Ebene gibt es entsprechende Bestrebungen. Durch deutsche und französische Akteure aus Politik und Wirtschaft wurde 2019 das Projekt Gaia-X [9] initiiert, in dem eine Dateninfrastruktur entstehen soll. Weiterhin gibt es durch die Europäische Kommission mit der European Open Science Cloud (EOSC) ähnliche Bestrebungen [18].

2.7.2 Architekturen

Die technische Ausgestaltung der Datenhaltung kann unterschiedliche Formen annehmen, welche jeweils mit Vor- und Nachteilen verbunden ist. Diese müssen zusammen mit den jeweiligen Anforderungen abgewogen werden. Im Folgenden werden verschiedenen Architekturen vorgestellt, anhand derer sich die geplante Architektur des Data Hubs orientiert.

Data Ecosystem

Bedingt durch die immer vielfältigeren Datenquellen und steigenden Datenmengen werden Strukturen benötigt, um diese Daten zu verwalten und zu verbreiten. Der Begriff des *Data Ecosystems* ist allerdings noch sehr neu und dementsprechend nicht einheitlich definiert [50, 65]. S. Oliveira, Barros Lima und Farias Lóscio geben in ihrer systematischen Literaturanalyse an, dass es sich grundsätzlich um ein soziotechnisches System handelt, in dem eine Infrastruktur für verschiedene Akteure mit unterschiedlichen Rollen

geschaffen wird [65]. Zentral dabei ist der Aspekt, dass Daten zum einen Dritten bereitgestellt werden können, zum anderen die Konsumenten der Daten ihre Ergebnisse aber auch wieder zurück in das System bringen können.

Dies unterstreicht zusätzlich eine der ersten Definitionen für Data Ecosystems von Pollock, in der das Data Ecosystem als kollaborative Plattform verstanden wird [57]. Der Plattformgedanke sieht hierbei vor, dass die Ergebnisse einer Datenverarbeitung wieder in der Plattform bereitgestellt werden. Dies zeigt den kollaborativen Gedanken eines Data Ecosystems auf, in dem die Zusammenarbeit verschiedener Akteure im Vordergrund steht. Dies ist im Kontrast zu klassischen Verbreitungswegen zu sehen, in dem Daten von einer Stelle bereitgestellt werden, von einer Weiteren verarbeitet werden und schließlich von einem Konsumenten genutzt werden.

Data Warehouse

Im Konzept des Data Warehouses werden verschiedene Datenquellen innerhalb eines neuen globalen Schemas integriert [47]. Die Daten werden aus den ursprünglichen Quellen in das Data Warehouse kopiert. Dabei können auch Transformationen stattfinden, die die Daten verändern und auf diese neue zentrale Ansicht anpassen. Dieser Prozess wird als Extract-Transform-Load (ETL) beschrieben. Zwar müssen die Daten dupliziert und im Rahmen der Transformation aufwändig integriert werden. Aber anschließend liegen die Daten genau in der benötigten Struktur vor. Dadurch lassen sich Abfragen entsprechend direkt auf den Daten ausführen. Allerdings bedeutet dies, dass die Daten nicht in ihrer ursprünglichen Form gespeichert werden und damit Details der Daten nicht mehr rekonstruiert werden können. Es handelt sich dabei um die sogenannte physische Integration [60].

Data Lake

Im Kontrast zum Data Warehouse gibt es das Konzept des *Data Lakes*. In einem Data Lake werden die Daten in ihrer ursprünglichen Form ohne weitere Verarbeitung oder Filterung gespeichert. Dies bedeutet, dass die Rohdaten immer in Gänze vorliegen und es keine Verluste, bedingt durch eine Transformation, gibt. Wird mit den Daten gearbeitet, begrenzt also kein zuvor definiertes Schema die Auswertung, da die Rohdaten in ihrer Gesamtheit untersucht werden können. Der Nachteil liegt aber nun darin, dass die Daten keinerlei globalen Struktur unterliegen (wie es beim Data Warehouse der Fall

ist) und damit immer während der Auswertung die diversen Eigenheiten der Rohdaten berücksichtigt werden müssen. Dies führt zu langsameren Auswertungen.

Bedingt durch die Idee des Data Lakes liegt auch ein Risiko in dieser Architektur. Sollte die Datenmenge zu groß werden, besteht die Gefahr eines *Data Swamps* [12]. Werden zu viele unstrukturierte Daten eingesammelt, kann dies das System überfordern und somit selbst kleine Auswertungen sehr kompliziert werden.

Data Cube

Der Data Cube ist vorrangig ein Bergriff, der mehrdimensionale Datenstrukturen beschreibt [30]. Informationen werden über verschiedene Achsen dargestellt, was die Interpretation der Daten erleichtern soll. Vor allem im Kontext von Unternehmen und Data-Warehousing finden sie Anwendung in Form von OLAP-Cubes. Hierbei können z. B. Kunden, Standorte und Verkäufe in einem Würfel betrachtet werden. Dies erlaubt es, die verschiedenen Dimensionen gleichzeitig zu betrachten und Fragestellungen wie „Wer ist der umsatzstärkste Kunde an einem bestimmten Standort?“ zu beantworten. Aber auch in anderen Domänen findet der Begriff Anwendung, wobei die Eigenschaft der Mehrdimensionalität ausgenutzt wird. Ein Beispiel ist die Erdbeobachtung [42], bei der ein Data Cube neben einer räumlichen Dimension zusätzlich den zeitlichen Verlauf eines Wertes an diesem Punkt beschreibt.

2.7.3 Geoinformationssystem

Ein Geoinformationssystem (GIS) liefert eine Umgebung für die Erhebung, Verarbeitung und Auswertung von räumlichen Daten [17]. Darin können verschiedene Datentypen wie Raster- oder Vektordaten untersucht werden. Zusätzlich ist auch eine Verschneidung der Daten mit weiteren, dynamischen Quellen möglich, wie Social Media Meldungen. Da viele Anwendungsszenarien einen räumlichen Bezug in der realen Welt haben, sind GISe essenziell. Darunter fallen bspw. Landnutzungsplanung, das Krisenmanagement bei Naturkatastrophen oder die langfristige Überwachung von Entwicklungen in Naturräumen.

2.8 Softwarelösungen

Es existiert eine große Zahl von verschiedenen Softwarelösungen im Themenbereich der GISe. Im Folgenden werden einige davon vorgestellt. Dabei werden Vor- und Nachteile der jeweiligen Systeme erläutert, um eine Einordnung des Data Hubs zu ermöglichen.

GeoNode GeoNode[15] ist ein Open Source „Content Management System (CMS) for geospatial data“ [51], das als Webanwendung funktioniert. Dabei können Datensätze mit Metadaten und interaktiven Karten für Dritte bereitgestellt werden. Zusätzlich können sie mittels spezifizierter Protokolle des Open Geospatial Consortium (OGC) abgerufen werden. Allerdings ist die Pflege der Metadaten noch sehr rudimentär, bspw. gibt es nur die Möglichkeit ein einziges Feld mit einem Zeitpunkt zu befüllen, aber nicht mehrere. Daher können keine individuellen Felder für Erhebungs- und Veröffentlichungszeitpunkt gesetzt werden. Außerdem steht kein Framework für die Verarbeitung von Daten zur Verfügung. Das System ist also – wie es selbst angibt – vor allem für die Veröffentlichung und Verteilung von Daten gedacht.

QGIS QGIS[16] ist eine plattformübergreifende GIS-Software als Desktopanwendung. Hierbei steht vor allem die Verarbeitung von Daten im Vordergrund. Verschiedene Datenquellen (u. a. GeoNode) können direkt in die Software integriert werden, aber auch lokale Dateien können eingelesen werden. Auf den Daten können entsprechende geografische Operationen zu Analysezwecken ausgeführt werden. Die Bedienung erfolgt aber vorrangig mittels der Benutzeroberfläche, wodurch eine umfangreiche Dokumentation nötig ist, wenn die Ergebnisse exakt reproduzierbar sein sollen. Zusätzlich bietet QGIS eine Plugin-Schnittstelle, mit der Erweiterungen für die Software entwickelt werden können. Das InaSAFE Plugin[17] bietet die Möglichkeit anhand verschiedener Daten ein Lagebild im Falle von Naturkatastrophen zu erstellen [59], mit dem Interventionen geplant und Entscheidungsträger informiert werden können. Allerdings ist das Plugin seit 2018 nicht mehr weiter entwickelt worden und bietet sich damit nicht zur weiteren Verwendung an. Außerdem ist es an die QGIS-Desktopsoftware gebunden.

[15] https://geonode.org/
[16] https://www.qgis.org/
[17] http://www.inasafe.org/home/index.html

Earth Map Earth Map[18] stellt eine freie, webbasierte Plattform zur Analyse großer Geodatensätze dar [46]. Die Website stellt dem Anwender verschiedene ausgewählte Datensätze bereit, die auf Basis von Landesgrenzen oder eigenen Area of Interests (AoIs) analysiert werden können. Die Verarbeitung der Datensätze findet in der Cloud der Google Earth Engine (GEE) statt. Eigene Datensätze können allerdings nicht integriert werden. Außerdem ist die Plattform zwar frei benutzbar, aber nicht Open Source. Bedingt durch die Anbindung an GEE wäre zudem eine lokale Installation und Benutzung nicht möglich. Dennoch steht durch die Kapazitäten der Cloud-Verarbeitung eine große Bandbreite verschiedener, umfangreicher Datenmengen zur Verfügung.

Keines der vorgestellten Werkzeuge erfüllt die Anforderungen (siehe Abschnitt 3.1) des Projektes vollständig. GeoNode liefert keine ausreichende Metadatenverwaltung oder Verarbeitungskapazitäten. In QGIS müssten Verarbeitungsroutinen umfangreich dokumentiert werden und zusätzlich müsste die Applikation für jeden Nutzer auf seinem Rechner installiert werden. Earth Map ist nicht vollständig Open Source und gebunden an die GEE und erlaubt zudem keine Integration eigener Datensätze. Zusätzlich ist auch keine klare Trennung der vorgestellten Architekturen möglich. Für den gewünschten Anwendungsfall müssen zum einen Rohdaten vorgehalten werden (Data Lake), zum anderen müssen sie aber auch integriert werden (Data Warehouse). Dies führte zur Entscheidung einer Eigenentwicklung, die auf die spezifischen Anforderungen angepasst ist.

[18] https://earthmap.org/

3 Methoden

In diesem Kapitel werden die Anforderungen an den Data Hub erläutert und anschließend die technische Umsetzung präsentiert.

3.1 Anforderungen

Die Anforderungen des Data Hubs ergeben sich vorrangig aus den Bedarfen im Rahmen des ESIDA-Projektes. Allerdings wird bereits bei der Konzeption über mögliche Erweiterungen und die Generalisierbarkeit des Systems nachgedacht. Dies ermöglicht neben der primären Funktion ein System, das bereits jetzt – oder zumindest mit planbaren Aufwänden – für andere Anwendungsszenarien eingesetzt werden kann. Dabei werden stets *Best Practices* der Softwareentwicklung berücksichtigt.

Openness (R1) Gesundheitsanwendungen sind eine kritische Infrastruktur, die Entscheidungsträgern oder Patienten wichtige Informationen bereitstellt. Diese Informationen können mitunter für sensible Entscheidungen verwendet werden. Daher ist es unabdingbar, dass die Software transparent im Umgang mit und in der Verarbeitung von Daten ist. Deshalb wird die Software unter der MIT-Lizenz entwickelt, welche die Open-Source-Kriterien erfüllt. Dies erlaubt es einem Anwender jederzeit den Quellcode zu untersuchen. Die MIT-Lizenz schließt zudem eine Entwicklung kommerzieller Produkte auf dieser Code-Basis nicht aus.

Nachvollziehbarkeit (R2) Die Entwicklung des Systems muss mit einem Versionskontrollsystem erfolgen (wie zum Beispiel git). Damit lässt sich eine transparente Kommunikation von Änderungen und Fehlerbehebungen abbilden. Zudem lässt sich anhand der einzelnen Änderungen eine Versionshistorie erstellen, die es Anwendern und interessierten Entwicklern ermöglicht, einen schnellen Überblick über das Projekt zu erhalten.

Reproduzierbarkeit (R3) Eng gekoppelt mit R1 ist die Reproduzierbarkeit. Durch den quelloffenen Ansatz ist dies zu einem gewissen Grad bereits erfüllt, allerdings gibt es in Anwendungen häufig Prozesse, die nur durch Interaktion des Anwenders durchgeführt werden können (bspw. der Ausschnitt von Karten oder Formatierung von Ergebnissen). Solche manuellen Schritte müssen im Data Hub grundsätzlich vermieden werden. Jeder *Schritt* in der Datenverarbeitung oder der Präsentation hat über den versionierten Quellcode zu erfolgen. Dies garantiert, dass die Ergebnisse jederzeit neu erstellt werden können.

On-premises (R4) Durch die potenzielle Verwendung von Gesundheits- und Patientendaten innerhalb des Systems, muss der Einsatz des Systems höchsten Sicherheitsstandards unterliegen. Zusätzlich muss der Anwender ein entsprechendes Vertrauen in die Software haben. Dies wird zum Teil bereits durch R1 erreicht. Allerdings muss auch gewährleistet sein, dass ein Anwender das System in seiner eigenen IT-Infrastruktur anwenden kann und nur berechtigten Nutzergruppen zugreifbar macht. Dies ist mit einem Cloud-Setup nicht realisierbar. Daher müssen getestete Anleitungen für das Self-Hosting bereitgestellt werden, sowie das System ohne unnötig komplexe Setups auskommen.

Wiederverwendbarkeit (R5) Das System wird mit Hinblick auf Tansania und das Denguefieber entwickelt. Allerdings sollen weitere Szenarien oder Einsatzgebiete ermöglicht werden. Daher soll das System generisch entwickelt werden, wodurch keine Kopplung an die konkreten Ausprägungen aus dem ESIDA-Projekt besteht.

3.2 Eingabedaten

Im Zuge von R1 ist erforderlich, dass die verwendeten Rohdaten ebenfalls der *openness* genügen. Daher müssen alle verwendeten Datenquellen unter offenen Lizenzen zur Verfügung stehen. Damit ein Anwender nachvollziehen kann, wie die Daten aufgebaut sind und wie sie verarbeitet wurden, ist es weiterhin nötig, die Grundinformationen der Datenquelle und der Verarbeitung kompakt einsehbar zu machen. Eine bloße Referenz auf die originale Quelle ist nicht ausreichend, da mitunter viele Quellen verwendet werden können. Im Kontext von ESIDA handelt es sich um 54 Quellen. Hierbei wäre ein schneller Überblick nicht mehr gewährleistet, wenn so viele Quellen selbst recherchiert werden müssten.

Dafür wurden zentrale Aspekte jeder Datenquelle, die *Metadaten*, innerhalb des Systems zusammengetragen und aufbereitet. In Tabelle 3.1 sind die Wichtigsten zusammengefasst. Für jede verwendete Datenquelle steht innerhalb des Systems eine kompakte Auflistung bereit. Dies erlaubt einen schnellen Überblick darüber, woher die Daten kommen und wie sie vorliegen. Dies dient als zentrales Repository innerhalb eines Projektes, um einen gemeinsamen Datenstand zu ermöglichen.

Tabelle 3.1: Verwendete Metadaten-Felder für eine Datenquelle.

Feld	Erklärung
Abkürzung	Eindeutige ID des Datums innerhalb des Systems.
Kategorie	Eine Kategorie, die die Quelle eingruppiert, wie *Wetter*, *Gesundheit*, *Infrastruktur*, ...
Titel	Beschreibender Titel
Notwendigkeit	Relevanz des Datums für das ESIDA-Projekt. `Required` oder `Optional`.
Format	Dateiformat, in dem die Daten vorliegen (CSV, TIFF, API, etc.).
Operation	Wie wurden die Rohdaten innerhalb des Systems aufbereitet (`min`, `max`, `mean`, `count`, ...)?
Originale Einheit	Wie liegen die Daten in der originalen Quelle vor?
Data Hub Einheit	Wie wurden die Daten aggregiert?
Räumliche Auflösung	Verwaltungsbereich, Koordinaten. Bei Rasterdaten, die Größe der Raster-Zellen (1 km, 5 km, ...).
Zeitliche Auflösung	Liegen die Daten bspw. täglich oder jährlich vor?
Zeitspanne	Für welchen Zeitraum liegen die Daten vor?
Source	Name/Beschreibung der Quelle der Daten.
Link	Online-Verweis auf die Quelle.
Zitierung	Empfohlene Zitierung durch die Quelle.
Sprache	Sprache der Quelle, in ISO 639-3.

Feld	Erklärung
Lizenz	Lizenz der Quelle, als SPDX-Abkürzung[1].
Abdeckung	Räumliche Abdeckung der Quelle (weltweit, Land, ...).

Die Metadaten werden innerhalb einer CSV-Datei im Projekt in Verbindung mit den Data Layern (siehe Abschnitt 3.4) gepflegt. Die Wahl fiel hier bewusst auf eine CSV-Datei und nicht etwa eine Datenbankstruktur. Dadurch ist zum einen die Wartbarkeit der Daten mit einfachen Mitteln gewährleistet (etwa Texteditor, oder Excel), zum anderen ist auch eine Versionierbarkeit (R2) der Daten in einem Klartext-Format ermöglicht. Dadurch kann in der Versionskontrolle immer der jeweilige Stand von Verarbeitung und Dokumentation synchron gehalten werden (R3). Somit gibt es keine Abhängigkeit zu manuellen Pflegetätigkeiten innerhalb eines datenbankgestützten Redaktionssystems.

3.2.1 Datenformate

Die identifizierten Datenquellen liegen in einer Vielzahl unterschiedlicher Formate vor. Dabei reicht die Bandbreite von einfachen CSV-Dateien, über komplexere Shapedateien oder GeoTIFF-Raster, bis hin zu Datenzugriffen, die nur mittels einer Application Programming Interface (API) stattfinden können. Die Formate lassen sich in die folgenden Kategorien einteilen, mit denen der Data Hub umgehen können muss.

CSV-Dateien

Die einfachste Form der Rohdaten liegt als CSV-Datei vor. Im Bereich der Gesundheitsforschung werden diese auch als *Line-List-Data* bezeichnet. Diese CSV-Dateien können beliebige Werte enthalten, wobei es in der Regel jeweils Spalten gibt, die einen temporalen und spatialen Bezug herstellen. Für den Import in den Data Hub können spezialisierte Mappings für jede Datenquelle angelegt werden. Für solche Daten bietet sich die Verarbeitung mittels der Python-Bibliothek Pandas [74] an. Hierbei kann mit einfachen Mitteln die Bereinigung, Filterung und bei Bedarf auch Aggregierung vorgenommen werden. Die Zuordnung zu den Regionen innerhalb des Data Hubs anhand der spatialen Eigenschaft findet dabei über einen Textvergleich der jeweiligen Bezeichnung der AoIs statt. Dabei

[1] https://spdx.org/licenses/

kann es vorkommen, dass in den verschiedenen Rohdaten unterschiedlichen Schreibweisen der Regionen verwendet werden. Im Falle der tansanischen Region Daressalam gibt es u. a. die alternativen Schreibweisen *Dar es Salaam* oder *Dar-es-Salaam*. Diese Unterschiede müssen beim Import geprüft werden. Der Data Hub bietet durch seine flexible Struktur einfache Möglichkeiten an, um solche Sonderfälle abzufangen.

Rasterdaten

Geografische Rasterdaten legen ein virtuelles Raster über eine Region. Dies kann lokal begrenzt oder weltumfassend sein. Das Raster hat dabei eine bestimmte Auflösung, die angibt, wie groß die reale Fläche der jeweiligen Zelle ist. Häufig verwendete Größen sind $1 \times 1\,\text{km}$ oder $5 \times 5\,\text{km}$. Bedingt durch die Krümmung der Erde können die real abgedeckten Flächen allerdings innerhalb eines Rasters leicht variieren. Jede Zelle hält einen spezifischen Wert. Es kann sich um eine Ganzzahl handeln, der bei Landklassifizierung eine Kategorie angibt, wie z. B. `1=Wald` oder `2=Urban`. Gleitkommazahlen als Zellenwert sind ebenfalls möglich, die etwa Werte wie eine Bevölkerungsdichte oder die durchschnittliche Höhe über dem Meeresspiegel beschreiben können. Eine Rasterdatei gibt in der Regel den Zustand zu einem bestimmten Zeitpunkt wieder. Um einen zeitlichen Verlauf abzubilden, wird üblicherweise für jeden Zeitschritt eine eigene Rasterdatei angelegt.

Rasterdaten können einen guten Überblick über die Datenlage in einem Land geben. Häufig werden sie auf Basis von Satellitenbildern modelliert und können damit Abschätzungen auch für abgelegene Gebiete liefern. Die Raster lassen sich ebenfalls auf generische Shapes abbilden (siehe Abschnitt 3.4.2). Von verschiedenen Quellen liegen umfangreiche Daten vor, wie Copernicus oder WorldPop.

Vektordaten

Vektordaten sind, wie Rasterdaten, raumbezogene Daten, die konkrete Punkte oder Linien beschreiben. Sie liegen in unterschiedlichen Dateiformaten vor, wie Shapefile oder GeoJSON. Die beschriebenen Punkte können bspw. Gesundheitseinrichtungen oder Mobilitätsknotenpunkte, wie Flughäfen oder Bahnhöfe, sein. Im Falle von Linien können sie Straßen oder die Trassierungen von Bahnstrecken beschreiben. Üblicherweise enthalten die Punkte oder Linien zusätzliche Informationen, die als generische Attribute ausgelesen und weiterverarbeitet werden können.

Programmierschnittstellen

Die anspruchsvollste Integration von Daten ist die von solchen, die mittels einer API angebunden werden müssen. Hierbei muss für jede API eine individuelle Anbindung entwickelt werden, die auf die jeweiligen Anforderungen und Abfrageregeln angepasst ist.

Meteostat Beispiel für eine API ist Meteostat[2], das aktuelle und historische Wetterdaten von Wetterstationen weltweit zusammenführt und bereitstellt. Mit der API ist es möglich, Wetterstationen innerhalb von bestimmten Ländern oder beliebigen geografischen Räumen abzufragen. Für jede Wetterstation können stündliche und tägliche Werte abgefragt werden. Es muss allerdings beachtet werden, dass nicht für jede Wetterstation und für jede angefragte Zeitspanne Daten zur Verfügung stehen. Die Abfrage kann über eine offizielle Python-Bibliothek durchgeführt werden, wie in Listing 3.1 gezeigt. Die Ergebnisse liegen direkt in einem Pandas-DataFrame vor.

Listing 3.1: Abfrage der Meteostat API mittels der offiziellen Python-Bibliothek.

```
# Import Meteostat library and dependencies
from datetime import datetime
from meteostat import Point, Daily

# Set time period
start = datetime(2018, 1, 1)
end = datetime(2018, 12, 31)

# Create Point for Hamburg, DE
hamburg = Point(53.551086, 9.993682)

# Get daily data for 2018
data = Daily(hamburg, start, end)
data = data.fetch()
```

STATcompiler Eine weitere API ist STATcompiler[3], über die Ergebnisse von Studien der Demographic and Health Surveys (DHS) Organisation abgefragt werden können. Die Studien umfassen Entwicklungsländer und erheben Gesundheits- und Bevölkerungsdaten, wie die Verbreitung der Nutzung von Moskitonetzen oder dem Bildungsstand der

[2] https://meteostat.net/
[3] https://www.statcompiler.com/

Bevölkerung. Die API erlaubt die Abfrage auf nationalem, sowie subnationalem Level. Zusätzlich bietet die API weitere Filter für spezifische Studien und Zeiträume[4]. Allerdings müssen die Anfrage-URLs selbst zusammengestellt werden und die Ergebnisse anschließend in einen DataFrame überführt werden, wie in Listing 3.2 zu sehen.

Listing 3.2: Abfrage der STATcompiler API mittels Request-URL.

```
import json
from urllib.parse import urlencode
from urllib.request import urlopen
import pandas as pd

# construct query and fetch data
params = {
    'countryIds':      'TZ',            # Country code
    'breakdown':       'national',      # national, or
       subnational
    'indicatorIds':    'ML_NETP_H_MOS', # Study ID
    'lang':            'en',
    'returnGeometry':  False,
    'surveyYearStart': 2010,
    'f':               'json',
}
data_url = 'https://api.dhsprogram.com/rest/dhs/data/?' +
   urlencode(params)

# Obtain and parse the result into a Python Object.
request = urlopen(data_url)
response = json.loads(request.read())

# load into Pandas Data Frame
df = pd.DataFrame(response['Data'])
```

OpenStreetMap OSM ist eine umfangreiche Quelle für Geodaten. Der Datenbestand wird von Freiwilligen gepflegt und kann für die Identifikation von Point of Interests (PoIs) oder Verkehrsnetzen wie Straßen oder Schienen genutzt werden. Dabei muss aber beachtet werden, dass die Daten sich kontinuierlich ändern oder falsch gepflegt, bzw. veraltet sein können, bedingt durch die offene Möglichkeit der Datenpflege durch Freiwillige. Dennoch bietet OSM eine gute Quelle für Infrastrukturdaten, die mittels der Python-Bibliothek OSMnx [10] abgefragt werden können, wie in Listing 3.3 demonstriert.

[4] https://api.dhsprogram.com/

Listing 3.3: Abfrage eines Schienennetzes von OSM mittels OSMnx.

```
import osmnx as ox
import geopandas

# polygon with the area of interest
shp = ...

# request tagged features from OSM
G = ox.graph_from_polygon(shp,
    simplify=True,
    retain_all=True,
    custom_filter='["railway"="rail"]'
)

# convert response to GeoDataFrame
gdf_nodes, gdf_edges = ox.graph_to_gdfs(G)
gdf_nodes = ox.io._stringify_nonnumeric_cols(gdf_nodes)
gdf_edges = ox.io._stringify_nonnumeric_cols(gdf_edges)
```

Wie die drei exemplarisch gezeigten APIs demonstrieren, ist ein Zugriff auf Daten mittels einer API immer abhängig von der jeweiligen Datenquelle und sehr spezialisiert. Bei der Datenintegration müssen damit immer spezifisch angepasste Wrapper erstellt werden, die diese Eigenheiten abstrahieren. Eine einmal eingerichtete Abstraktion kann allerdings auch sehr mächtig sein, da sie mehrfach eingesetzt werden kann. So erlaubt die Abfragelogik für DHS den direkten Zugriff und die Integration von jeder der über 400 Studien.

3.3 Area of Interest

Die Eingabedaten liegen in unterschiedlichen räumlichen Bezügen vor, bspw. auf administrativen Einheiten, als geografisches Raster, oder nur Punkt-/Vektordaten. Um eine Vergleich- und Nutzbarkeit der Daten herzustellen, müssen sie auf einen gemeinsamen geografischen „Nenner“ gebracht werden. Dies wird im Data Hub durch die AoIs, bzw. Shapes abgebildet. Dafür wurde eine generische, hierarchische Struktur entwickelt, in der geografische Gebiete gespeichert werden können. Kern dieser Struktur ist die in Abb. 3.1 dargestellte Klasse. Ein Shape umfasst immer eine Geometrie (`geometry`), wie etwa ein Rechteck oder Polygon, zusammen mit einem Namen (`name`) und eventuell vorhandenen weiteren Attributen in einer flexiblen Key-Value-Struktur als JSON

(`properties`). Zusätzlich kann eine Eltern-Kind-Relation über die Kombination von `id` und `parent_id` zwischen einzelnen Shapes hergestellt werden.

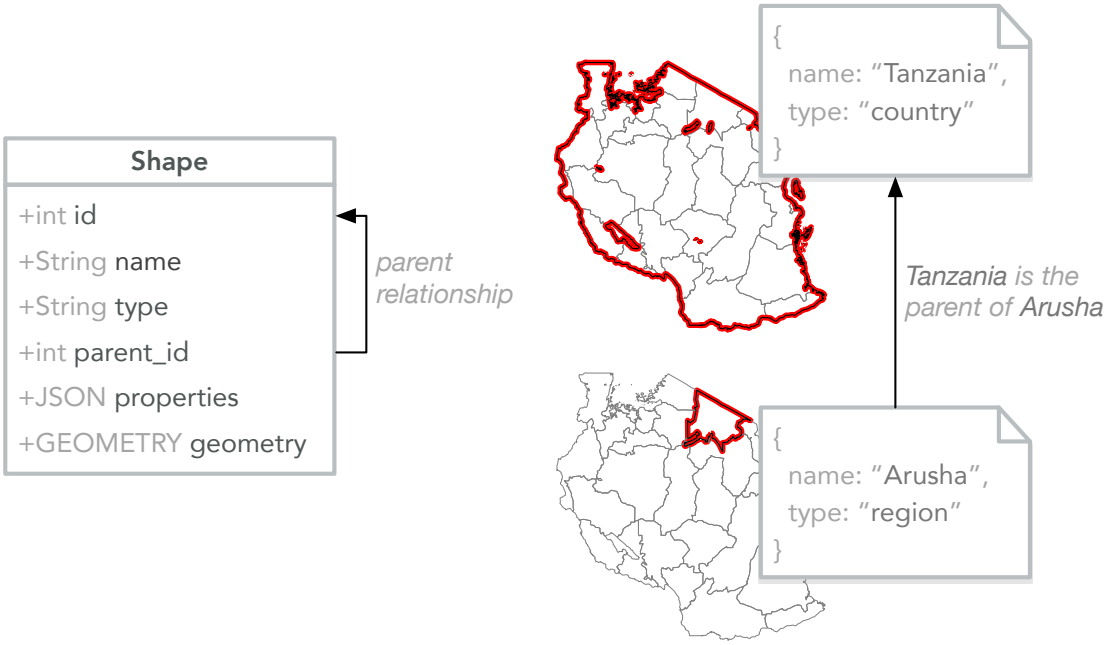


Abbildung 3.1: Struktur der Shapes im Data Hub.

Hierbei können beliebige geografische Räume in einen Zusammenhang gebracht werden. Es kann dementsprechend nicht nur ein einzelnes AoI im System vorgehalten werden, sondern beliebig viele. Das `type`-Attribut ermöglicht es, verschiedene *Kategorien* von geografischen Räumen anzulegen, wie die hierarchischen Verwaltungseinheiten eines Landes (Land, Bundesland, Landkreis) oder generische Strukturen, wie Rechtecke oder beliebige Polygone.

Die Struktur des AoI gibt damit technisch keinerlei Anforderungen vor. Es muss sich lediglich um eine geschlossene, valide Geometrie handeln, wobei Daten im Koordinatenreferenzsystem EPSG:4236 kodiert sein müssen[5]. Diese können als Shapefile (`.shp`) oder GeoPackage (`.gpkg`) mit dem in Listing 3.4 gezeigten Command-line-interface (CLI)-Aufruf in das System eingelesen werden.

[5]Technisch sind andere Koordinatenreferenzsystem möglich, im Rahmen des Projektes bot dieses Referenzsystem allerdings die größte Schnittmenge zwischen benötigten Daten und Shapefiles.

Listing 3.4: Import von geografischen Räumen in den Data Hub.

```
1 python ./esida-cli.py load-shapes FILE
```

In vielen Bereichen sind die administrativen Verwaltungseinheiten eine übliche Auflösungstiefe für die Kommunikation von Daten. Dies sind Länder und deren Unterteilungen. Vor allem in der Epidemiologie findet dies so statt, da Fallzahlen üblicherweise auf der Ebene von Verwaltungseinheiten aggregiert und übermittelt werden. Des Weiteren werden viele Informationen auf Basis von Verwaltungseinheiten gesammelt, wodurch diese Daten dann auch mit Bezug zu diesen veröffentlicht werden (so zum Beispiel bei DHS). Die Behörden der jeweiligen Länder stellen in der Regel entsprechende Shapefiles zur Verfügung, wie zum Beispiel im Open Data Portal Kanadas[6], der USA[7], oder Deutschlands[8]. Relevant für das ESIDA-Projekt waren die Verwaltungseinheiten von Tansania, welche ebenfalls durch eine tansanische Behörde als Shapefile zum offenen Download[9] bereitgestellt werden.

Sollten für das benötigte Gebiet keine offiziellen Shapefiles auffindbar sein, kann auf andere Quellen zurückgegriffen werden. Das Projekt geoBoundaries[10] stellt Shapefiles für alle Länder und deren verschiedene administrative Ebenen zusammen [64]. In diesem Projekt sind zudem Vergleiche zwischen verschiedenen Quellen und abweichenden Umrissen der Shapes möglich. Außerdem stellt die Weltbank einen Datensatz mit den Grenzen aller Länder der Welt[11] bereit, welcher frei verwendet werden kann und häufig als Grundlage für Publikationen in unterschiedlichsten Domänen dient.

Neben administrativen Gebieten als AoIs innerhalb des Data Hubs können aufgrund der generischen Struktur auch beliebig gebildete Geometrien eingelesen werden. Es können beispielsweise schachbrettartige Strukturen verwendet werden, wie sie bei Rasterdaten zugrunde liegen. Mittels des Geohash-Algorithmus[12] lassen sich solche Rechtecke auch hierarchisch strukturieren. Durch den Technologiekonzern Uber wurde ein ähnliches Sys-

[6] https://open.canada.ca/en
[7] https://data.gov/
[8] https://basemap.de/
[9] https://www.nbs.go.tz/index.php/en/census-surveys/gis/568-tanzania-districts-shapefiles-2019
[10] https://www.geoboundaries.org/
[11] https://datacatalog.worldbank.org/search/dataset/0038272
[12] Der Autor Gustavo Niemeyer hat den Algorithmus 2008 gemeinfrei auf Wikipedia veröffentlicht, um eine Patentierung zu verhindern: https://en.wikipedia.org/wiki/Geohash

tem auf Basis von Hexagonen anstelle von Rechtecken erstellt, mit dem Namen H3[13]. Der Vorteil von Hexagonen besteht darin, das die Mittelpunkte benachbarter Zellen immer gleich weit voneinander entfernt sind.

3.4 Data Layer

Der *Data Layer* kombiniert die konkrete Information einer Datenquelle mit der jeweiligen AoI. Damit wird es möglich, die Informationen sowohl auf der räumlichen Ebene, zugeschnitten auf die AoIs, als auch auf der zeitlichen Ebene zu betrachten und abzufragen. Damit entsteht ein dem Data Cube (siehe Abschnitt 2.7.2) ähnliches Gebilde, das es ermöglicht, die Daten anhand verschiedener Dimensionen zu betrachten (räumlich und zeitlich). Der Data Layer bietet zusätzlich die Möglichkeit, die zuvor identifizierten Metadaten abzufragen.

Ein Data Layer enthält immer genau eine Metrik. Das bedeutet, dass für eine Datenquelle mehrere Data Layer existieren können, wenn unterschiedliche Informationen in der Quelle enthalten sind. Im Falle der Copernicus Landnutzungsdaten etwa gibt es für jede Kategorie (Waldfläche, Urban, ...) einen entsprechenden Data Layer. Für Wetterdaten von Meteostat bspw. gibt es für jede Metrik, wie Niederschlag oder Temperatur, einen Data Layer. Die Metrik wird immer in Bezug zum jeweiligen AoI und der zeitlichen Auflösung der Datenquelle aufbereitet.

Jeder Data Layer bekommt einen eindeutigen Namen, mittels welchem er innerhalb des Data Hubs identifiziert werden kann. Der Name kann die Zeichen `[a-z0-9_]+` enthalten, wobei die entwickelte Konvention die Struktur `<Datenquelle>_<Metrik>` empfiehlt. Dadurch ergeben sich Data Layer Namen wie `copernicus_forest` für Waldflächen aus den Copernicus Landnutzungsdaten oder `meteo_tprecit` für Niederschlagsmengen aus den Wetterdaten von Meteostat. Für jeden Data Layer gibt es eine entsprechende Quellcode-Datei mit gleichen Namen, in der die Klasse des Data Layers mit diesem Namen definiert wird.

Technische Grundlage ist ein objektorientiertes System verschiedener Klassen, mit dem die unterschiedlichen Typen von Eingabedaten abgebildet werden können (siehe Abschnitt 3.2.1). Generische Funktionen werden in einem `BaseParameter` bereitgestellt, und können je nach spezifischer Art der Daten angepasst oder ergänzt werden.

[13] https://h3geo.org/

3.4.1 Implementierung

Alle Data Layer leiten sich von dem `BaseParameter` ab, in dem die gemeinsamen Grundfunktionalitäten gebündelt sind. Dabei handelt es sich um interne Hilfsfunktionen zum Downloaden von Dateien oder dem Persistieren der Ergebnisse innerhalb der Datenbank. Zwischen konkreter Data-Layer-Klasse und dem `BaseParameter` können weitere Abstraktionsschichten existieren, die Funktionalitäten von spezialisierten Datenquellen vereinen, wie zum Beispiel weitere Hilfsfunktionen für die Verarbeitung von Rasterdaten (`TiffParameter`) oder API-Zugriffe (`DHSParameter`, `MeteostatParameter`). Diese Struktur folgt dem Ansatz der Objektorientierte Programmierung (OOP) und ist damit leicht erweiterbar für neue Datenquellen oder Funktionen. Ein Überblick dieser Struktur ist in Abb. 3.2 gegeben.

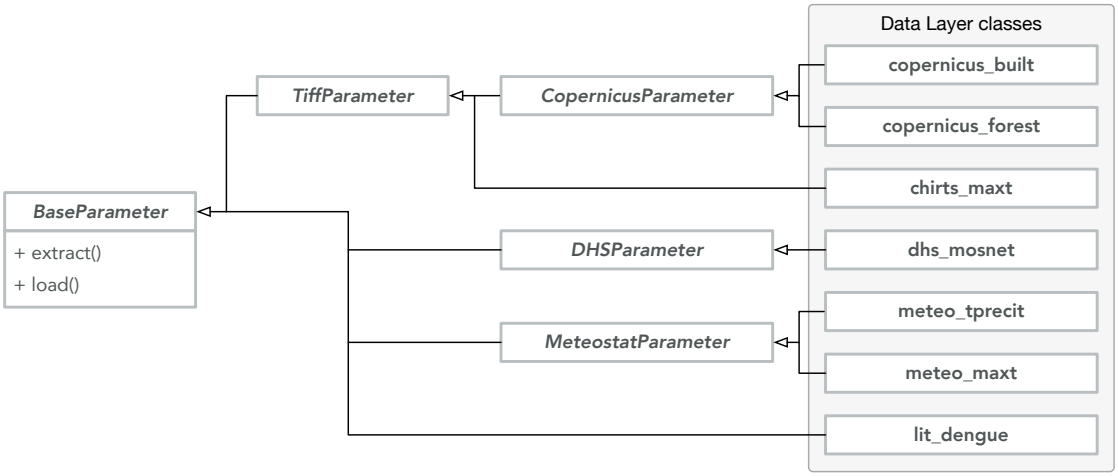


Abbildung 3.2: Vererbungshierarchie der verschiedenen Data-Layer-Klassen.

Der `BaseParameter` definiert eine `extract()` Methode, die für das Extrahieren, bzw. Downloaden, der Rohdaten aus der jeweiligen Datenquelle zuständig ist. In der konkreten Data-Layer-Klasse wird die Methode entsprechend der Anforderungen der Datenquelle übersteuert. In der Methode `load()` definiert jeder Data Layer selbst die benötigten Transformationen und Berechnungen zum Ermitteln der gewünschten Metrik, wobei auch hier auf geteilte Hilfsfunktionen zurückgegriffen werden kann. Gerade bei API-getriebenen Integrationen bietet sich die Vereinheitlichung an. Im Falle der DHS Data Layer wird bspw. in der abstrakten Klasse `DHSParameter` die API-Logik implementiert, sodass in den konkreten Klassen lediglich die gewünschte Studie angegeben

werden muss und dann vollautomatisch integriert wird. Diese Integration der Daten, über im Quellcode definierte Regeln, ist ein wichtiger Baustein für das Erfüllen von Anforderung R3 (Reproduzierbarkeit).

Die beiden Methoden dienen als globale Einhängepunkte, durch welche die Integration der Daten global gesteuert werden kann. Der Data Hub stellt eine CLI bereit, mit der alle, bzw. individuelle, Data Layer heruntergeladen und integriert werden können. Die CLI kann entweder manuell genutzt werden oder in automatisierte und zeitgesteuerte Systeme integriert werden. Damit lassen sich zum Beispiel wiederholte Importe von APIs realisieren, die in verschiedenen Abständen neue Daten enthalten und damit einen *kontinuierlichen* Fluss von aktuellen Daten in den Data Hub ermöglichen.

3.4.2 Datenintegration

Die beiden Methoden `extract()` und `load()` für die Datenintegration innerhalb des Data Layers bilden damit einen (vereinfachten) ETL-Prozess ab, wobei die *Transformation* innerhalb der beiden Schritte stattfindet. Diese Vereinfachung entlastet bei der Integration den Entwickler, da keine genaue Trennung in drei Teile vorgenommen werden muss. Durch die Fokussierung auf die Bereiche „Download“ und „Verarbeitung“ findet eine klare Struktur Anwendung, was zu einem vereinfachten mentalen Modell des Systems und damit zur Usability beiträgt.

Die Daten können dabei von den folgenden Quellen integriert werden:

- Wenn die Quelle keine (nutzbare) Schnittstelle anbietet, werden die Daten innerhalb des Projektes abgelegt. Gründe hierfür können sein, dass die Datenquelle eine Zugriffsbeschränkung/Authentifizierung einsetzt, der Server keine zuverlässige Verfügbarkeit anbietet oder es zu häufigen Änderungen am Portal kommt. Außerdem können selbst zusammengestellte Daten so integriert werden.
- Download der Rohdaten von den Servern der Datenquelle, wie bspw. FTP-Server oder OGC-kompatibler Server, mit anschließendem Vorbereiten für die Verarbeitung.
- Zugriff mittels API, um direkt die benötigten Daten abzufragen.

Die Integration der Daten findet sukzessiv auf jededem dem System bekannten AoI statt. Dabei werden die Rohdaten jeweils mit der Geometrie des Shapes verschnitten und mit

der Logik innerhalb der `load()`-Methode verarbeitet. Für die individuellen Anforderungen eines Data Layers liefert die `load()`-Methode entsprechende Verarbeitungsmöglichkeiten im Zusammenspiel von Rohdaten und dem spezifischen geografischen Raum. Dies umfasst u. a. die folgenden Möglichkeiten:

`count` Für Vektordaten ist eine einfache Operation, wie das Zählen bestimmter PoIs innerhalb des Gebietes, möglich. Konkret könnte dies eine Liste von Gesundheitseinrichtungen in einem Land sein, deren Anzahl je Region gezählt werden soll, um so einen Eindruck der Verteilung zu erhalten.

`mean` Für Rasterdaten können Zellen beispielsweise einen Messwert je Zelle enthalten. Für den gesuchten geografischen Raum kann der Durchschnitt aller im Gebiet enthaltenen Zellen berechnet werden.

`proportion` Manche Rasterdaten enthalten eine Klassifizierung je nach Zelle, bspw. ob es sich um bewaldete oder urbane Gebiete handelt. Der Prozess ermittelt die Gesamtanzahl der Zellen, die von dem Raum abgedeckt werden und ermittelt den Anteil der gesuchten Klassifizierung innerhalb des Gebietes.

Darüber hinaus erlaubt das flexible System spezialisierte Verarbeitungen je nach Anforderungen des Data Layers. Diese können durch den Anwender in den jeweiligen `load()`-Methoden implementiert werden.

Nachdem der Prozess durchgelaufen ist, werden die errechneten Daten in einer PostGIS-Datenbank persistiert. Der Prozess speichert für jeden Wert den temporalen und räumlichen Bezug. Der Durchlauf der Integration ist dabei deterministisch und idempotent.

Da damit der komplette Prozess von Download → Aufbereitung → Verarbeitung der Versionskontrolle unterliegt und vollständig mittels Quellcode realisiert werden **muss**, sind die Anforderungen R2 und R3 sichergestellt. Es kann damit, systemisch, keine manuellen Arbeitsschritte geben, die nicht dokumentiert oder für die Endergebnisse relevant sind. Dieses Vorgehen lässt sich mit dem Leitsatz *configuration through code* beschreiben. Durch den Wegfall von manuellen Schritten ist es jederzeit möglich, das System neu auszuführen und Ergebnisse zu reproduzieren (unter der Bedingung, dass die Eingabedaten nicht verändert wurden).

3.4.3 Datenabfrage

Die berechneten Daten können vom Data Layer auf unterschiedlichen Wegen abgefragt werden. Hierfür steht ein Download im CSV- oder Excel-Format zur Verfügung. Diese einfachste Form des Zugriffs stellt die aggregierten Daten auf zeitlicher und räumlicher Dimension dar, die CSV- und Excel-Dateien ermöglichen dabei eine schnelle und einfache Möglichkeit für manuelle Analysen in einem umfangreichen Ökosystem (R, Python, SPSS, ...). In Abb. 3.3 ist ein derartiger Export exemplarisch dargestellt. Die Spalte `data_layer` enthält die konkreten ermittelten Werte des Data Layers, die für das jeweilige Gebiet und den Zeitpunkt gültig sind. Im realen Anwendungsfall würde diese Spalte als Namen die ID des Data Layer tragen.

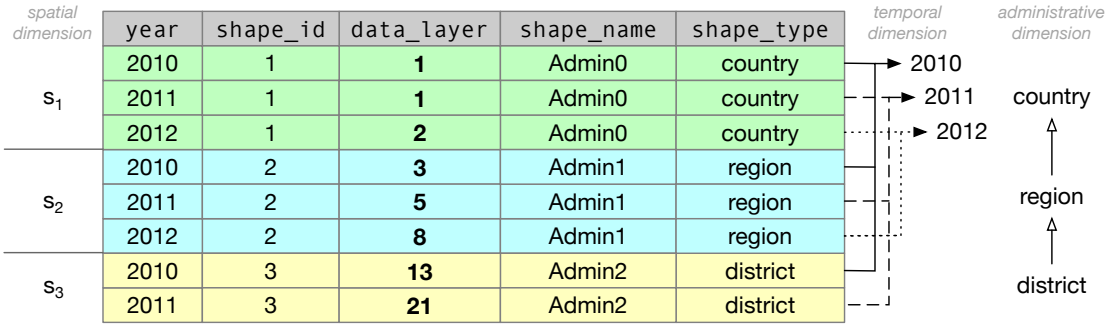


Abbildung 3.3: Exemplarischer Export eines Data Layers als CSV-Datei.

Die CSV-Datei bietet sowohl Möglichkeiten der temporalen Analyse für verschiedene Shapes, als auch für den räumlichen Überblick zu einem bestimmten Zeitpunkt. Hervorzuheben ist, dass in Abb. 3.3 exemplarisch für die Shape s_3 die Information im Jahr 2012 nicht vorhanden ist, was auf fehlende Rohdaten für diese Region und Zeit zurückzuführen ist. Daher gibt es keinen entsprechenden Eintrag in der CSV-Datei für das Tupel (s_3, 2012, `data_layer`).

Neben dem Download steht zusätzlich der Zugriff mittels einer REST-API zur Verfügung, durch welche die Daten programmatisch in der jeweiligen Analyse-Software eingelesen werden und mit weiteren Data Layern verschnitten werden können. Dies ermöglicht eine mächtige Schnittstelle zur Auswertung der spatio-temporalen Daten.

Zusätzlich können die Daten interaktiv im Webinterface des Data Hubs eingesehen und mit einfachen Mitteln, wie Heatmaps oder Liniendiagrammen, aufbereitet werden, um einen ersten Überblick der Daten zu erhalten, wie in Abb. 3.4 zu sehen. Die Daten

eines jeweiligen Layers können in einer jeweiligen räumlichen Auflösung als Heatmap für einen bestimmten Zeitpunkt dargestellt werden, bspw. Regions-Ebene zum Zeitpunkt 2020. Zusätzlich kann der zeitliche Verlauf für eine bestimmte Region als Liniendiagramm dargestellt werden, im Beispiel der Abbildung *Daressalaam*.

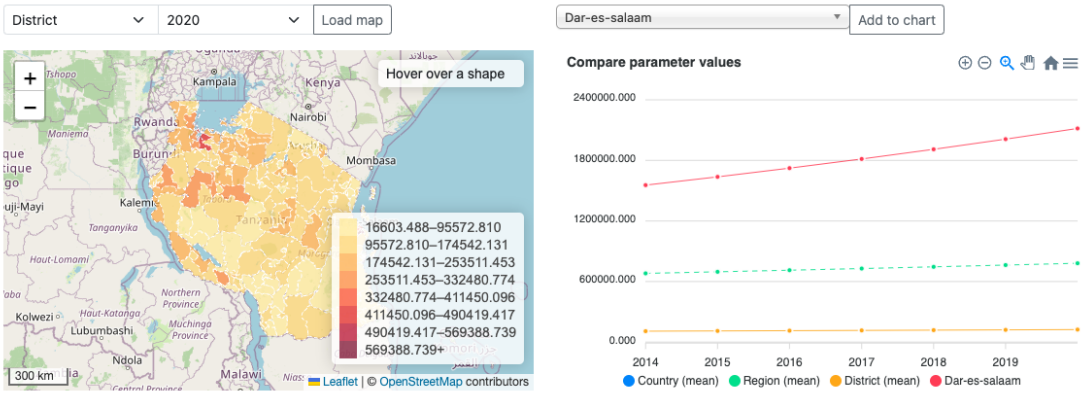


Abbildung 3.4: Analysemöglichkeit eines Data Layers im Data Hub.

Zur Beurteilung eines aktuellen Signals, wie einer Krankheitsmeldung, erlaubt der Data Hub zusätzlich die aggregierte Einsicht verschiedener Data Layer für ein bestimmtes Shape. Dabei werden jeweils die letzten bekannten Werte der Data Layer angezeigt. Mittels dieser Übersicht kann ein Benutzer sich schnell einen Überblick der Datenlage der verschiedenen Data Layer einholen und die Situation des gegebenen Signals beurteilen (siehe Abb. A.3).

Wie in dem Beispiel der CSV-Datei (siehe Abb. 3.3) zuvor bereits angedeutet, ist es nicht gewährleistet, dass für jede Region zu jedem Zeitpunkt ein Wert ermittelt werden kann. Um mit diesen *Lücken* umgehen zu können, finden verschiedene, konfigurierbare Fallbacks statt. In Abb. 3.5 ist der gesamte Algorithmus zur Abfrage eines Wertes für eine bestimmte Region und Zeitpunkt schematisch dargestellt. Hierbei wird zuerst überprüft, ob ein Wert für die Kombination der Anfrageparameter existiert. In diesem Fall kann der Algorithmus abbrechen und den gefundenen Wert zurückliefern. Ist dies nicht der Fall, wird geprüft, ob auf der temporalen Achse der Werte für dieses Shape ältere oder neuere Werte bekannt sind. Je nach Konfiguration wird dann ein entsprechender Wert zurückgegeben. Im Standardverhalten wird der ältere Wert zurückgegeben, da dieser aus Sicht des Anfragezeitpunktes zumindest der zuletzt bekannte Wert ist.

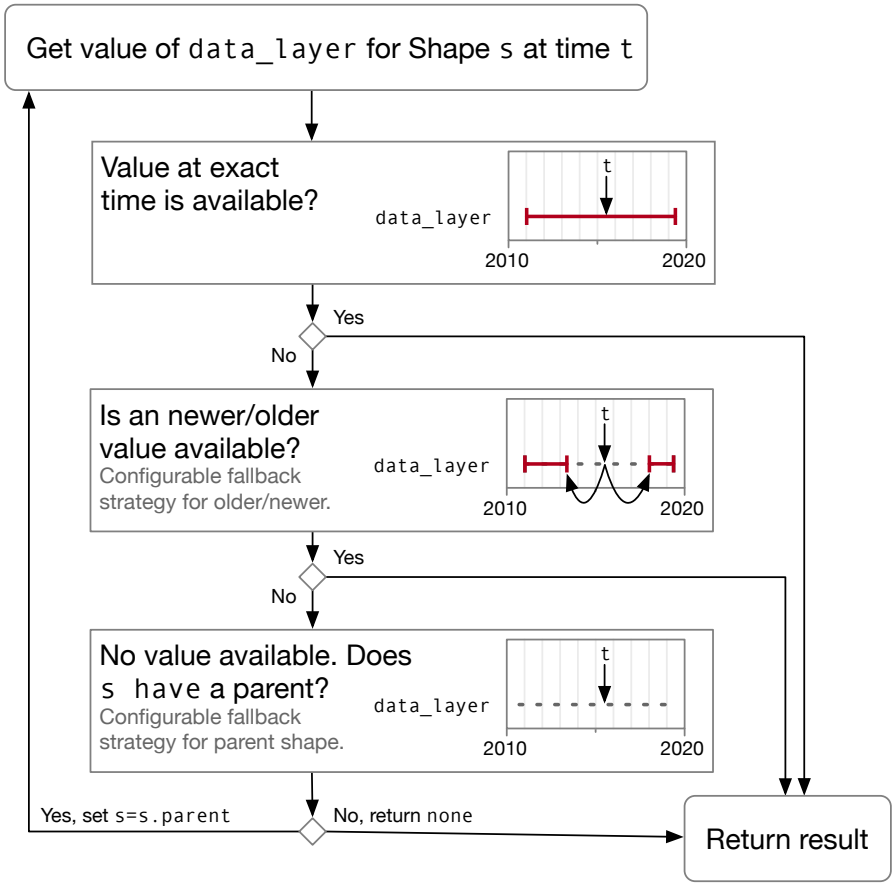


Abbildung 3.5: Ablauf Werte-Abfrage eines Data Layers im Rahmen eines Algorithmus.

Ist innerhalb des Data Layers, für die angefragte Region, kein Wert verfügbar, wird überprüft, ob die Region in einer Eltern-Kind-Relation zu einem anderen Shape steht. Ist dies der Fall, wird der Algorithmus die Anfrage erneut ausführen, allerdings für das Eltern-Shape. Damit besteht die Möglichkeit, im Falle von fehlenden Daten zumindest Daten mit einer gröberen Auflösung einer höheren geografischen Instanz verwenden zu können, bspw. den Wert für ein Land, anstelle des angefragten Bundeslandes. Dieser Fallback wird zusätzlich in den heruntergeladenen Dateien, bzw. im Webinterface, kenntlich gemacht. Der Anwender kann somit selbst entscheiden, ob die Daten *noch gut genug* sind oder ob er eine zusätzliche Datenquelle recherchieren und integrieren muss. Im obigen CSV-Beispiel in Abb. 3.3, in dem für Shape s_3 (`district`) kein Wert im Jahr 2012 bekannt ist, würde der Algorithmus entsprechend den Wert des übergeordneten Shapes s_2 (`region`) zurückgeben, was im Jahr 2012 der Wert 8 ist.

3.5 Data Hub

Der *Data Hub* bildet den Überbau für die einzelnen Data Layer. Wie die vorherigen Teile ist das System in Python implementiert. Als Grundlage dient das Microframework Flask[14]. Der Data Hub stellt Strukturen bereit, durch die sowohl mit AoIs als auch mit den Data Layern individuell interagiert werden kann und damit die Verbindung zwischen diesen ermöglicht wird. Das System setzt sich aus zwei zentralen Komponenten zusammen: Dem Data Hub selbst, bestehend aus dem Python Quellcode, und einer Datenbank für die Persistierung der Daten. In Abb. 3.6 ist der Überblick schematisch dargestellt.

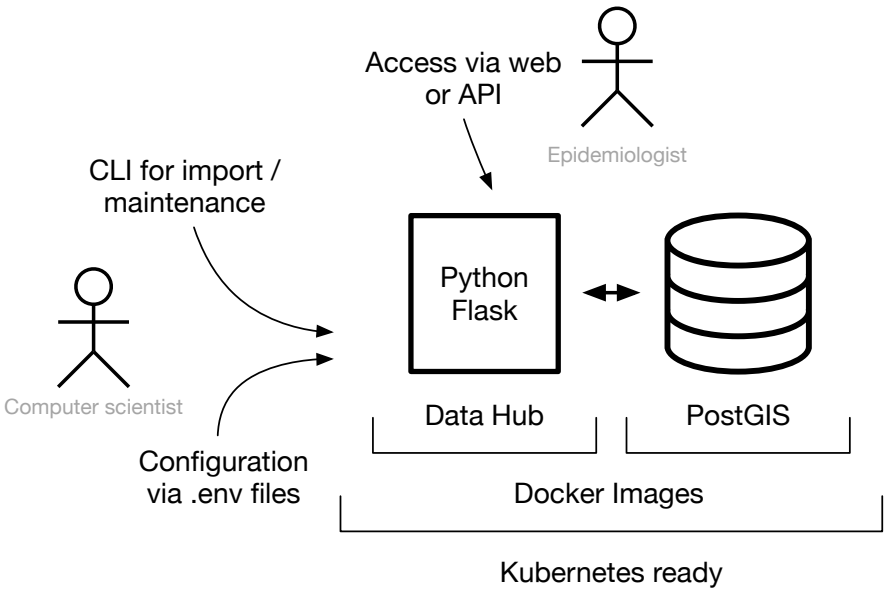


Abbildung 3.6: Schematischer Überblick der Technik des Data Hubs.

Die Wahl fiel auf PostGIS als Datenbank, da sie als Geodatenbank optimierte Funktionen für die Speicherung und Abfrage von räumlichen Daten anbietet. Da PostGIS eine Erweiterung der relationalen Datenbank PostgreSQL ist, stehen allerdings auch klassische Operationen einer relationalen Datenbank zur Verfügung. Damit können die Ergebnisse der Datenprozessierung gut gespeichert und abgefragt werden.

Durch die Grundlage des Flask-Frameworks können des Weiteren ohne großen Aufwand Webansichten sowie die REST-API entwickelt werden. Flask bietet als Microframework nur einen strukturellen Rahmen und dennoch viele grundlegende Funktionen für eine

[14] https://flask.palletsprojects.com/

Webapplikation, steht aber nicht durch hohe Komplexität einer schnellen Entwicklung im Weg. Umfangreiche Funktionen, wie eine Benutzerverwaltung, gibt es damit allerdings nicht. Wäre dagegen ein vollwertiges Webframework, wie Django, verwendet worden, wäre die Einarbeitung für die Entwicklung deutlich komplexer und zeitaufwändiger gewesen. Flask ist in Python entwickelt, was sich ebenfalls anbietet, da in Python bereits ein umfangreiches Ökosystem zur Verarbeitung verschiedener Daten besteht (Pandas [74]). Vor allem existieren auch umfangreiche Bibliotheken zur Verarbeitung von Geodaten (Rasterio [29]) oder der Abfrage von OSM-Daten (OSMnx [10]). Dadurch kann sowohl die Verarbeitung der Daten als auch die Präsentation im Web mit derselben Sprache realisiert werden, was zur Harmonisierung der Entwicklung beiträgt.

Die Gesamtapplikation ist innerhalb von Docker-Containern zusammengefasst. Dies erlaubt die schnelle Installation auf verschiedenen Plattformen. Docker stellt eine Virtualisierungsumgebung bereit, in dem alle Abhängigkeiten des Projektes enthalten sind. Daher sind keine umfangreichen Installationsanleitungen nötig und Probleme mit Inkompatibilitäten von Softwarepaketen unwahrscheinlich. Alle Versionen sind fest hinterlegt und aufeinander abgestimmt. Das Setup wurde unter macOS entwickelt und sowohl im Kubernetes-Cluster der Hochschule für Angewandte Wissenschaften Hamburg (HAW Hamburg), als auch auf einem bereitgestellten Server des Bernhard-Nocht-Institut für Tropenmedizin (BNITM) installiert; damit ist es unter unterschiedlichen Hardware-Architekturen und System-Konstellationen erprobt.

3.5.1 Softwarearchitektur

Den Kern der Architektur des Data Hubs bildet ein Python-Modul, das die verschiedenen Komponenten verbindet. In Abb. 3.7 ist ein Überblick über das System gegeben. Das System setzt sich vorrangig aus drei Bereichen zusammen. Die Entwicklung findet primär durch die Erstellung und Erweiterung der Data-Layer-Klasse, wie bereits zuvor erläutert, statt. Hier liegt zudem die Verarbeitung der gesammelten Rohdaten, welche auf dem Dateisystem gespeichert sind. Die Administration des Systems erfolgt mittels einer CLI und der Konfiguration über Umgebungsvariablen. Die CLI erlaubt die Installation des Systems, die Interaktion mit den Data Layern (Download und Integration), sowie Import- und Export-Funktionen. Mittels der Umgebungsvariablen können hartkodierte Einstellung, wie für den Datenbankzugriff, vermieden werden und individuell gesteuert werden. Im letzten Bereich finden sich die an den Benutzer gerichteten Komponenten. Dies beinhaltet die HTML-Templates für das Frontend, die API und das Routing.

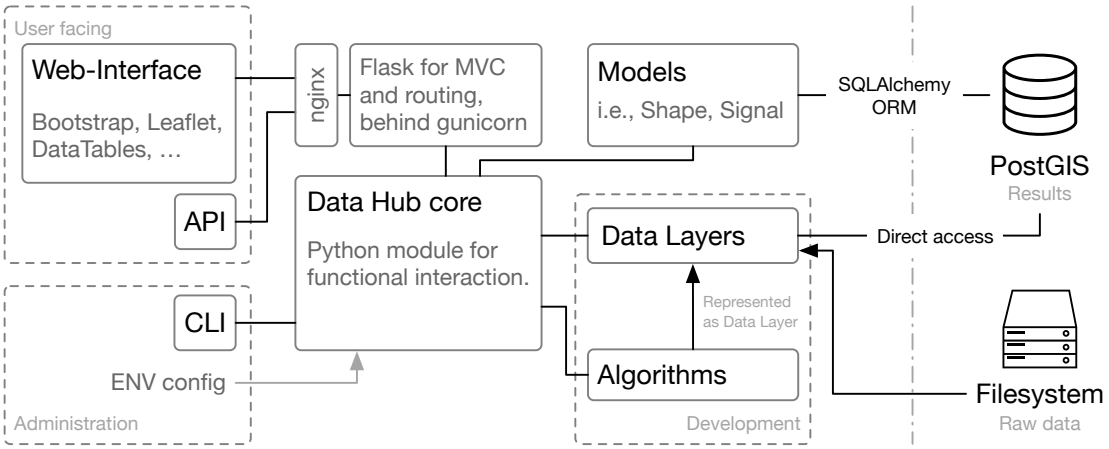


Abbildung 3.7: Übersicht des Zusammenspiels der Komponenten im Data Hub.

Das Frontend wird, entsprechend der Auslieferung an den Browser, in HTML, CSS und JavaScript entwickelt. Als Basis dient dabei das Frontend-Framework Bootstrap[15]. Es stellt einzelne, aufeinander abgestimmte Bausteine zur Verfügung, mit denen ein homogen gestalteter Auftritt entworfen werden kann. Für die interaktiven Elemente wird für die Tabellen das DataTables[16] Framework verwendet, durch das mittels *Progressive enhancement* einfache HTML-Tabellen filterbar und sortierbar gemacht werden können. Für die Erstellung von interaktiv bedienbaren Karten wurde auf Leaflet[17] zurückgegriffen. Die dynamischen Diagramme wurden unter anderem mittels Plotly[18] realisiert.

Um das Frontend an den Browser auszuliefern wird im Backend das Flask-Microframework genutzt. Dies realisiert das Routing, in dem die unterschiedlichen URLs den entsprechenden Funktionsbausteinen zugeordnet werden. Dies folgt dem Model-View-Controller (MVC)-Pattern. Mittels des Routings ist zusätzlich die REST-API realisiert. Für die Authentifizierung der Anwender ist die gesamte Applikation hinter einen nginx Proxy geschaltet. In diesem Proxy wird die Anmeldung mittels HTTP BasicAuth durchgeführt, was bedeutet, dass es innerhalb der Applikation keine Rollen- und Rechteverwaltung gibt.

Der Zugriff auf die Datenbank erfolgt mittels SQLAlchemy[19], dabei handelt sich um ein Object–Relational-Mapping (ORM)-Framework. Die Modelle (Shape, Signal) wer-

[15] https://getbootstrap.com/
[16] https://datatables.net/
[17] https://leafletjs.com/
[18] https://plotly.com/
[19] https://www.sqlalchemy.org/

den durch das ORM-Framework definiert und das entstandene Modell wird durch das Framework in ein entsprechendes Datenbank-Schema überführt und angelegt. Zudem findet die Abfrage und Selektion von Einträgen über das ORM statt. Dadurch reduziert sich die Menge von Code und es muss keine händische SQL-Abfrage für die Datenbank geschrieben werden, was die Entwicklungsgeschwindigkeit und Sicherheit erhöht. Im Fall von großen Datenmengen nach der Prozessierung durch die Data-Layer-Logik bietet sich allerdings kein ORM an. Durch den Overhead des ORMs würde hier der Nutzen durch eine stark reduzierte Performance aufgehoben werden. Für diese Daten erfolgt der Zugriff durch manuell geschriebene Abfragen oder direkt mittels Funktionen aus der Pandas-Bibliothek, die für DataFrames entsprechende Datenbank-Wrapper mit hoher Performance anbietet.

Die Pandas-Bibliothek bildet auch einen zentralen Aspekt bei der internen Verarbeitung der Daten. Sollten durch Benutzeranfragen verschiedene Data Layer gemeinsam abgefragt werden, müssen diese auf der räumlichen, bzw. zeitlichen, Achse vereint werden. Da es sich hierbei um große Tabellen handelt, wird dabei auf Funktionen der Bibliothek zurückgegriffen. Sie bietet sich dafür an, da spezielle, optimierte „merge“-Operationen dafür existieren.

3.5.2 Designentscheidungen

In den folgenden Abschnitten werden einige Elemente bzw. Herangehensweisen des Data Hubs detaillierter beleuchtet.

Verarbeitung der TIFFs

Bei den TIFF-Rasterdaten handelt sich teilweise um sehr große Dateien. Bspw. sind die TIFF-Dateien des Data Layers für die Alters- und Geschlechterverteilung (Data Layer IDs `worldpop_sexage_*`) jeweils ca. 423 MB groß. Unabhängig vom benötigten Ausschnitt der Datei, muss immer zuerst die gesamte Datei eingelesen werden, was eine umfangreiche I/O-Operation ist. Würde für jedes Shape einzeln die Integrations-Berechnung durchgeführt und dabei jeweils die TIFF-Datei neu eingelesen werden müssen, würde dies zu langen Ausführungszeiten führen. Dieses Problem wurde umgangen, indem jede TIFF-Datei nur einmal in den Speicher eingelesen wird und dann anschließend für jedes benötigte Shape eine Maske erstellt wird. Mit dieser Maske wird auf dem im Speicher

gehaltenen TIFF die benötigte Berechnung durchgeführt. Dies reduziert die Anzahl der benötigten Zugriffe auf das Dateisystem immens.

Release-Prozess

Durch eine größere Nutzerbasis im ESIDA-Projekt, sowie grundsätzlich durch die Veröffentlichung als Open Source, entsteht der Bedarf, Änderungen, Funktionen und Inkompatibilitäten neuer Versionen der Software zu kommunizieren. Das Änderungsprotokoll der Versionsverwaltung gibt hier zwar schon Aufschluss über die Änderungen, allerdings ist dieser nicht kompakt, und kann durch eine hohe Zahl von Änderungen, die keine konkrete Kommunikation an den Anwender bedürfen, schnell unübersichtlich werden. Deshalb hat sich in vielen Projekten die Pflege eines Changelogs[20] etabliert, in dem alle nennenswerte Änderungen zwischen zwei Versionen dokumentiert werden, sowie Zeitpunkt und Versionsnummer der neuen Version.

Im Falle des Data Hubs ist diese Pflege in Form eines Release-Prozesses automatisiert, der sich auf die Commit-Messages der Versionskontrolle stützt. Dafür wird das Konzept der Conventional Commits[21] verwendet. Hierbei werden Commits in einem entsprechenden Format beschrieben, das Rückschlüsse auf die Bedeutung zulässt. Eine neue Funktion erhält bspw. das Präfix `feat:`, eine Verbesserung der Dokumentation `docs:`. Das Konzept des Semantic Versioning[22] [20] sieht vor, dass eine Versionsnummer dem Schema `<major>.<minor>.<patch>` folgt. Dabei wird die `<major>`-Nummer im Falle von Inkompatibilitäten (*Breaking change*) inkrementiert, `<minor>` und `<patch>` bei neuen Funktionen bzw. Bugfixes. In der Kombination von Conventional Commits und Semantic Versioning kann automatisiert anhand der Historie der Commit-Messages die neue Versionsnummer bestimmt werden, sowie der Changelog vorausgefüllt werden. Ein so kodifizierter Prozess erleichtert sowohl die Entwicklung durch die Automatisierung des Prozesses, als auch die Möglichkeit der Anwender, die Weiterentwicklung des Projektes nachvollziehen zu können.

[20] https://keepachangelog.com/
[21] https://www.conventionalcommits.org/
[22] https://semver.org/

Komplexität

Hohe Komplexität einer Software kann zu höheren Fehlerquoten und geringerer Produktivität der Entwickler führen [72]. Allerdings kann ein *einfaches* System unzureichend flexibel sein, um benötigte Funktionen abzubilden. Zwischen diesen beiden Extremen muss immer vorsichtig abgewogen werden. Im Falle des Data Hubs lag der Fokus darauf, Komplexitäten, wo möglich, zu reduzieren. Daher wurde die Benutzerauthentifizierung bspw. aus dem System herausgelöst und lediglich über den nginx-Proxy realisiert. Dadurch ist keine Implementierung oder Berücksichtigung von Zugriffsrechten im System nötig – andererseits bedeutet das, dass es keine rollenspezifischen Berechtigungen geben kann.

Ein weiterer Punkt, in dem die Komplexität bewusst reduziert wurde, findet sich im Frontend. Ein moderner Softwarestack der Webentwicklung umfasst eine Vielzahl verschiedener Tools. Dieses Ökosystem unterliegt zudem einer hohen Fluktuation in Hinblick auf die Beständigkeit von Frameworks. Um hier die Wartbarkeit des Systems zu erhöhen, wurde bewusst auf komplexe Build-Prozesse verzichtet. Dadurch werden JavaScript Dateien einzeln an den Browser ausgeliefert und nicht konkateniert und minifiziert, was zu höheren Ladezeiten führen kann. Im Kontext einer Software, die nicht in Konkurrenz zu anderen Produkten steht – wie ein Online-Shop – sind dies aber akzeptable Abstriche, die dafür die Komplexität stark reduzieren.

Dogfooding

Der Data Hub stellt die API nicht nur Dritten für Abfragen bereit, sondern nutzt diese auch selbst, um Daten im Kontext des Webinterfaces zu visualisieren. Dies findet bei den dynamischen Diagrammen statt, die der Anwender selbst zusammenstellen kann, in Bezug auf temporale und räumliche Ausdehnung (bspw. die Heatmaps). Die Technik, die eigenen Produkte auch selbst zu verwenden, wie etwa in einer Software die API nicht nur für Dritte anzubieten, sondern zusätzlich selbst für Funktionen zu nutzen, wird als *dogfooding* bezeichnet [33]. Die Bezeichnung geht auf eine amerikanische Fernsehwerbung der 1980er zurück, in der ein Tierfuttermittelhändler seinem Hund das selbst verkaufte Futter verfütterte („*eating your own dog food*“). Der Mehrwert dieser Herangehensweise ist, dass die eventuellen Fehler in der externen API bereits in der internen Entwicklung auffallen, sowie die API nicht nur abstrakt bereitgestellt wird, sondern auch von den

Entwicklern selbst *gelebt* wird und damit unintuitive Situation bei der Verwendung der API direkt auffallen.

3.6 Workflow-Engine

Ein weiterer Baustein des Data Hubs bildet die Workflow-Engine. Mit dieser können auf Basis der vorhandenen Data Layer Algorithmen definiert werden, die die vorhandenen Daten verarbeiten. Dabei handelt es sich technisch um einen abstrakten Data Layer, implementiert in der `AlgorithmParameter`-Klasse. Damit gliedert sich diese Funktion in die hierarchische Struktur der Data Layer ein und profitiert von den bereits etablierten Grundfunktionen. Speicherung, Exploration und Download der Ergebnisse stehen damit analog zu *normalen* Data Layern direkt zur Verfügung - ohne weiteren Entwicklungsaufwand. Da es sich wiederum um einen Data Layer handelt, folgt hierbei auch wieder eine komplette Reproduzierbarkeit, wie in der Anforderung R3 beschrieben.

Grundlage dafür ist eine Konfigurationsdatei, in der die zu verwendenden Data Layer beschrieben werden und *was* mit den abgefragten Werten geschehen soll. Die Abfrage iteriert dabei über jedes geladene Shape und führt den Algorithmus aus. Die Konfigurationsdatei wird mittels YAML[23] erstellt. Diese YAML-Dateien werden innerhalb der Versionskontrolle getrackt, damit die Nachvollziehbarkeit (R2) gewährleistet bleibt.

In Listing 3.5 ist exemplarisch ein Auszug einer solchen Konfiguration gezeigt[24]. Im Zeitraum zwischen dem 1.1.2010 und 1.12.2019 wird in monatlichen Abständen für jedes Shape der Data Layer `copernicus_built` abgefragt und in Abhängigkeit des Anteils der urbanen Fläche innerhalb des Shapes ein Score von 1, 2 oder 3 vergeben.

[23] Die Wahl fiel hierbei auf YAML, da sie für Konfigurationsdateien zum einen weitverbreitet ist, wie bei Docker oder Kubernetes, und zusätzlich Kommentare erlaubt, im Gegensatz zu JSON.

[24] Vollständige Konfiguration: https://github.com/MARS-Group-HAW/esida-db/blob/main/input/algorithms/esida-risk-assessment-likelihood.yaml

Listing 3.5: Auszug aus einer YAML-Konfiguration eines Algorithmus.

```yaml
apiVersion: v1
metadata:
  name: Local Risk-Assessment
  range:
    interval: MS      # monthly intervalls
    start: 2010-01-01
    end: 2019-12-01
spec:
  - name: Urban environment / built-up %
    datalayer: copernicus_built
    mode: percentage
    thresholds:
      - max: 10
        inclusive: False
        score: 1
      - min: 10
        max: 50
        score: 2
      - min: 50
        inclusive: False
        score: 3
```

Der schematische Ablauf ist in Abb. 3.8 zu sehen. Für jeden Zeitschritt (t) wird für jedes verfügbare Shape ($s_1, \ldots$) der Algorithmus durchgeführt. Dabei wird zuerst für jeden definierten Data Layer der entsprechende Wert abgefragt, der für den Zeitpunkt t im Shape s_n gilt. Dies geschieht mittels der zuvor vorgestellten Abfrage im Data Layer (Abschnitt 3.4.3). Anschließend wird der Rückgabewert entsprechend der Vorgaben des Algorithmus (σ) ausgewertet. Im aktuellen Stand handelt es sich dabei um die Abfrage der Grenzwertbereiche und die Vergabe eines entsprechenden Scores. Abschließend wird jeder Score mittels der Aggregationsfunktion (τ) zum Endergebnis berechnet, in diesem Fall eine Summe.

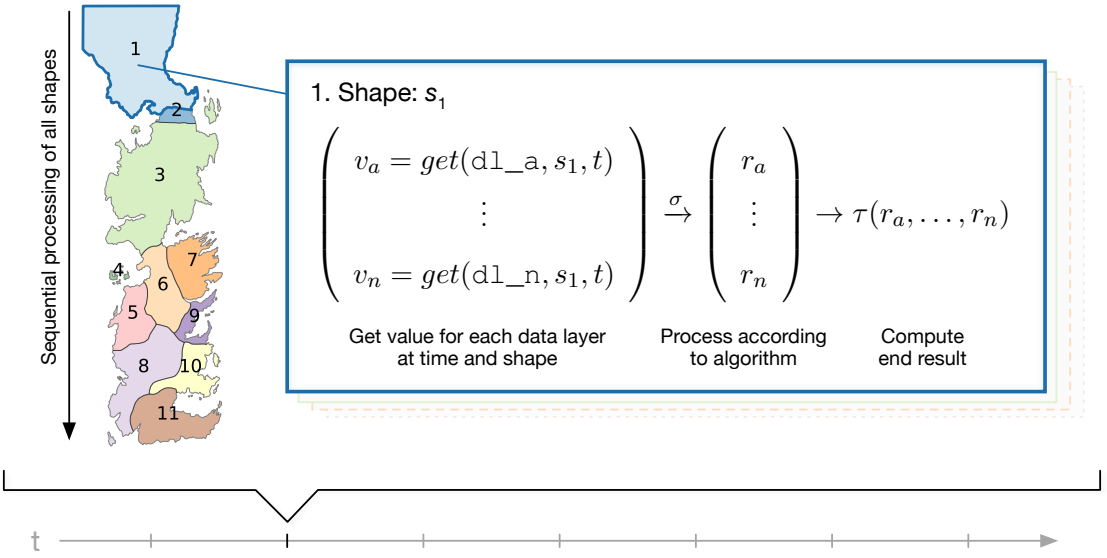


Abbildung 3.8: Generischer Ablauf eines Algorithmus im Data Hub. Quelle des Kartenmaterials theMountainGoat[a], CC BY-NC-SA 3.0.

[a] https://www.cartographersguild.com/showthread.php?t=30472

Bedingt durch die Bedarfe des Projektes sind zum aktuellen Zeitpunkt für mögliche Algorithmen σ nur die Abfrage in Form von Grenzwerten vorgesehen und für die Endergebnisse mittels τ nur die Summenbildung. Allerdings bieten die Data Layer durch die flexible Struktur der Vererbung, sowie die Möglichkeiten der Konfigurationsdateien, Einstiegspunkte, um neue Funktionen hinzuzufügen.

Für die Ergebnisse eines Algorithmus wird, wie eingangs erwähnt, die Struktur des Data Layers verwendet, um direkt die Visualisierungsmöglichkeiten des Data Layers in räumlicher und zeitlicher Dimension nutzen zu können. Damit einhergehend stehen die Download- und API-Funktionen ebenfalls zur Verfügung, was die Weiternutzung der Ergebnisse in angeschlossenen Workflows erlaubt. Zudem stehen für die „Algorithmus Data Layer“ umfangreiche Logs zur Verfügung, die im Webinterface eingesehen werden können. Dies erlaubt eine Nachvollziehbarkeit der Ergebnisse, was einerseits das Debugging erleichtert, anderseits aber auch Vertrauen in die Ergebnisse erhöht, da der Anwender direkt nachvollziehen kann, wie bestimmte Ergebnisse sich zusammensetzen.

3.7 Signal-Bewertung

Für einzelne Krankheitsmeldungen, bzw. Signale, steht eine Eingabemaske zur Verfügung. In dieser können die Metadaten zu einem Fall eingetragen werden, der bspw. aus der Event-based Surveillance bekannt wurde. Der Fall kann mit verschiedenen Metainformationen, vor allem Zeitpunkt und Ort, im Data Hub hinterlegt werden. Anhand dieser Merkmale kann ein konfigurierbarer Bewertungsalgorithmus angestoßen werden, der auf Basis des Ortes die Shapes identifiziert, in denen der geografische Punkt liegt. Für diese Shapes wird ein Abfragealgorithmus ausgeführt, in dem ausgewählte Data Layer abgefragt werden und eine Einschätzung des Signals vorgenommen wird. Die Konfiguration erfolgt wie bei der zuvor erläuterten Workflow-Engine mittels YAML-Konfigurationsdateien.

4 Ergebnisse

In diesem Kapitel werden die Ergebnisse vorgestellt. Die Grundlage stellt dabei die im vorangegangenen Kapitel erläuterte Architektur. Informiert wurden die Datenquellen und Anforderungen aus dem Kontext des ESIDA-Projektes.

4.1 Quellcode

Der Data Hub ist unter der MIT-Lizenz auf GitHub veröffentlicht[1]. Entsprechend der Anforderungen und der vorgestellten Architektur handelt es sich dabei nicht nur um die Quelldateien des Systems, sondern schließt auch die Konfiguration der Datenquellen und Algorithmen ein. Durch die Installation des Projektes kann damit die Gesamtheit der Ergebnisse reproduziert werden. Da die gesamte Architektur innerhalb von Docker-Containern gebündelt ist, lässt sich der Data Hub mit geringem Aufwand auf unterschiedlichen Systemen starten. Auf dem Kubernetes-Cluster der HAW Hamburg lief das System bis zum Angriff auf die IT-Infrastruktur[2] der HAW Hamburg im Dezember 2022. Im Zuge des Angriffs wurde der Kubernetes-Cluster abgeschaltet, was zu einem Verlust der Installation führte.

Durch die flexible Struktur über Docker konnte das System allerdings schnell lokal auf den jeweiligen Computern der Anwender erneut installiert werden. Durch die realisierte Import- und Export-Funktion konnten die Ergebnisse der Datenintegration und Algorithmen auch ohne den direkten Zugriff auf die Rohdaten verteilt werden. Dadurch war es nicht erforderlich für die einzelnen Benutzer die langwierigen Downloads abwarten zu müssen und große Datenmengen auf dem eigenen Computer abzuspeichern (siehe Abschnitt 4.2). Somit konnte *nahtlos* mit den Ergebnissen weiter gearbeitet werden.

[1] https://github.com/MARS-Group-HAW/esida-db
[2] https://www.haw-hamburg.de/cyberangriff/

Zusätzlich konnte das System auf einem vom BNITM bereitgestellten Server mittels Docker installiert werden, wodurch das System wieder unabhängig von lokalen Installationen allen Projektbeteiligten zur Verfügung steht. Diese verschiedenen Installationen haben die Reproduzierbarkeitsmerkmale zum einen validiert und zum anderen auch gezeigt, wie notwendig sie sind.

Die zuvor in Abschnitt 2.6 vorgestellten Regeln für reproduzierbare Ausführung von Programmcodes wurden, soweit sinnvoll, eingehalten. Zusätzlich hängen diese Regeln eng mit den abgeleitete Anforderungen R1, R2 und R3 zusammen:

- Die Regeln 1 (*For Every Result, Keep Track of How It Was Produced*) und 2 (*Avoid Manual Data Manipulation*) ergeben sich direkt aus den genannten drei Anforderungen.
- Regel 3 (*Exact Versions of All External Programs Used*) ergibt sich aus der Verwendung und Bereitstellung eines Docker-Containers, indem die Versionen aller verwendeter Software definiert sind.
- Regel 4 (*Version Control*) ergibt sich aus R2 (Nachvollziehbarkeit), indem eine Versionskontrolle eingefordert wird.
- Regel 5 (*Record All Intermediate Results*) wurde eingehalten, indem die Data Layer und Algorithmen Logdateien anlegen, die zum Debugging dienen.
- Regel 6 (*Note Underlying Random Seeds*) wurde nicht explizit eingehalten, da keine Funktionen mit Zufall vorliegen. Allerdings könnten Seeds über die etablierten Strukturen zur Eingabe von Umgebungsvariablen gesteuert werden.
- Regel 7 (*Always Store Raw Data behind Plots*) ist eingehalten, da Visualisierungen auf Basis der Daten in der Datenbank erstellt werden und diese vorliegt.
- Regel 8 (*Generate Hierarchical Analysis Output*) ist eingehalten, da die Weiterverarbeitung, bspw. in Algorithmen, auf anderen Data Layern aufbaut, und die Data Layer selbst immer in der Datenbank persistiert sind.
- Regel 9 (*Connect Textual Statements to Underlying Results*) ist nicht anwendbar.
- Regel 10 (*Provide Public Access*) ist erfüllt durch Anforderung R1 (Openness). Der Quellcode ist, wie zuvor erläutert, öffentlich einsehbar und unter MIT lizenziert.

4.2 Daten

In den Data Hub sind 110 Data Layer integriert worden. Diese leiten sich von 54 verschiedenen offenen Datenquellen ab. Die Größe der lokal abgelegten Rohdaten beläuft sich

dabei auf ca. 238,4 GB. Die Data Layer setzen sich thematisch aus fünf Kategorien zusammen (Demografie, Umwelt, Gesundheit, Infrastruktur und Wetter), wie in Abb. 4.1 zu sehen. Davon sind 36 Data Layer zwingend für die Durchführung des Risiko-Algorithmus (siehe Abschnitt 4.3) erforderlich. Die Weiteren bieten zusätzlichen Kontext oder Alternativen an. Die benötigten Data Layer sind in Tabellen A.1 und A.2 aufgeführt, wobei in Tabelle A.1 die Metadaten und Beschreibungen enthalten sind und in Tabelle A.2 die jeweiligen Quellen der Data Layer.

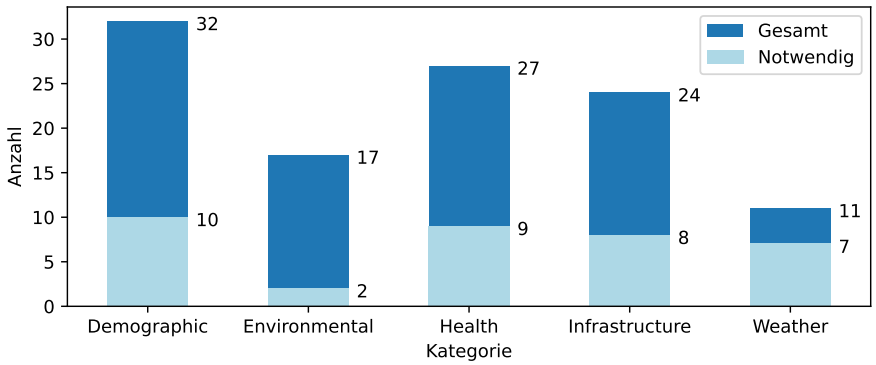


Abbildung 4.1: Anzahl aller und notwendiger Data Layer je Kategorie im Data Hub.

Der angestrebte zeitliche Rahmen innerhalb des ESIDA-Projektes wurde auf 2010–2020 definiert, da in dieser Zeitspanne mehrere Denguefieber-Ausbrüche in Tansania aufgetreten sind, mit dem größten Ausbruch in 2019 [48]. Somit kann der Ansatz für diese Zeitspanne an belegten Ausbrüchen validiert werden. Für die benötigten Data Layer ergibt sich entweder eine volle Abdeckung oder mindestens eine partielle Abdeckung der Zeitspanne, wie in Abb. 4.2 zu sehen. Die abgebildete Spanne gibt je den ersten und letzten Datenpunkt an und berücksichtigt daher eventuelle Lücken innerhalb der Zeitpanne nicht. Dies kann aber durch die angegebene Vollständigkeit erkannt werden.

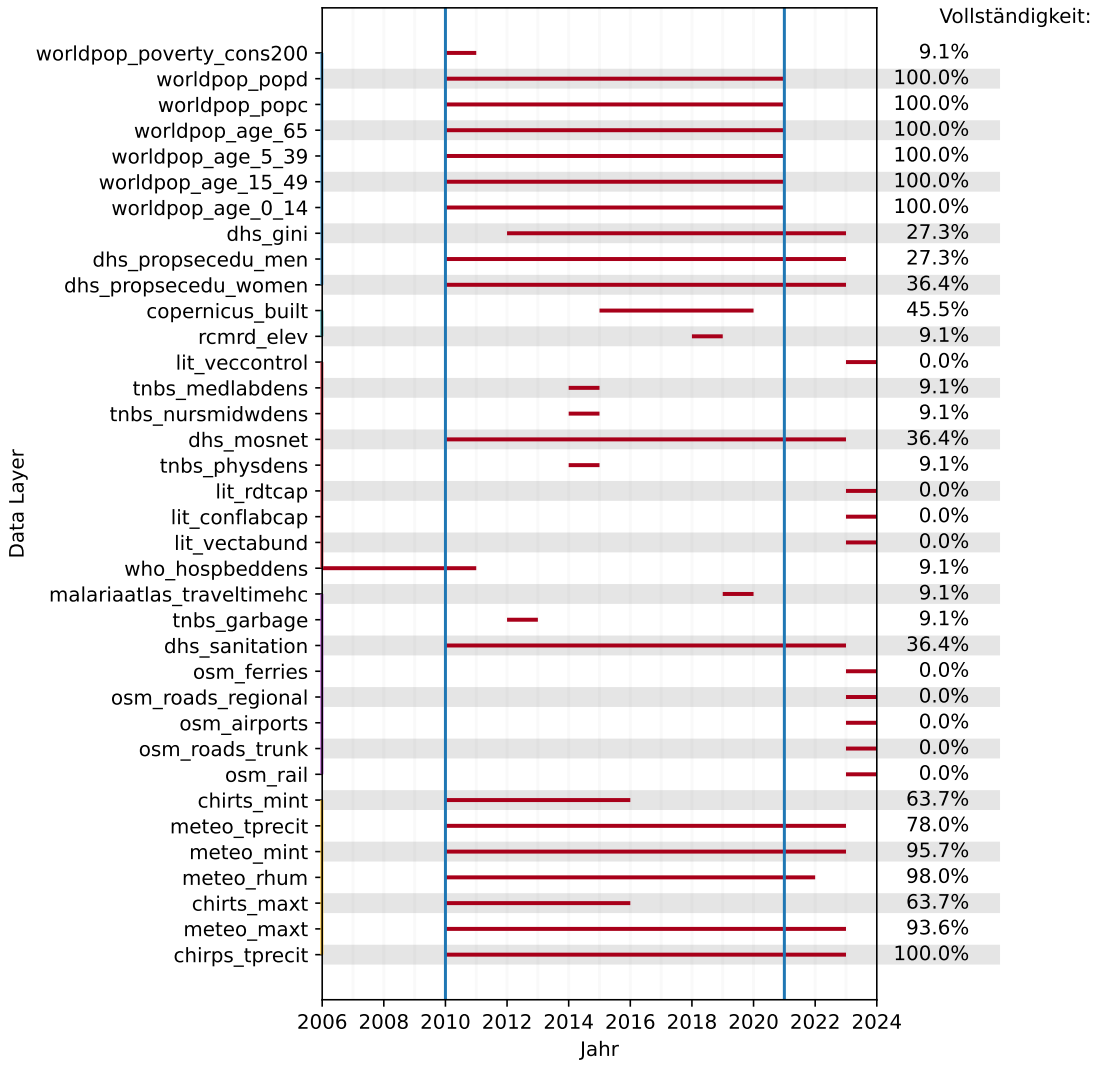


Abbildung 4.2: Temporale Vollständigkeit der benötigten Data Layer im Intervall 2010–2020.

Dabei zeigt sich, dass für die Demografie- und Wetterdaten eine nahezu vollständige Abdeckung erreicht werden konnte. Wobei durch die periodische, meist mehrjährige, Erhebung durch Studien oder Zensus große Lücken bestehen. Im Falle der Infrastrukturdaten, die zu großen Teilen von OSM integriert wurden, besteht die geringste Abdeckung, da diese nur für den aktuellen Zeitraum abgefragt werden können. Wobei die Infrastruktur sachgemäß einer deutlich geringeren Varianz unterliegt, als Bevölkerung oder Wetter.

Für die spatiale Dimension wurden Shapefiles der tansanischen Statistikbehörde[3] verwendet. Durch diese konnte die im ESIDA-Projekt definierte, minimale spatiale Auflösung von *Districts* abgebildet werden. Zusätzlich konnte die übergeordnete Verwaltungseinheit der *Regions* eingelesen werden und übergeordnet das Land selbst (*Country*). In Abb. 4.3 sind die jeweiligen Auflösungen und ihre Geometrien dargestellt. In Summe konnten damit 227 AoIs angelegt werden, die sich auf die drei Kategorien aufteilen. Zwischen den Shapes besteht jeweils eine Eltern-Kind-Relation, die die Hierarchie Country ⇒ Region ⇒ District abbildet.

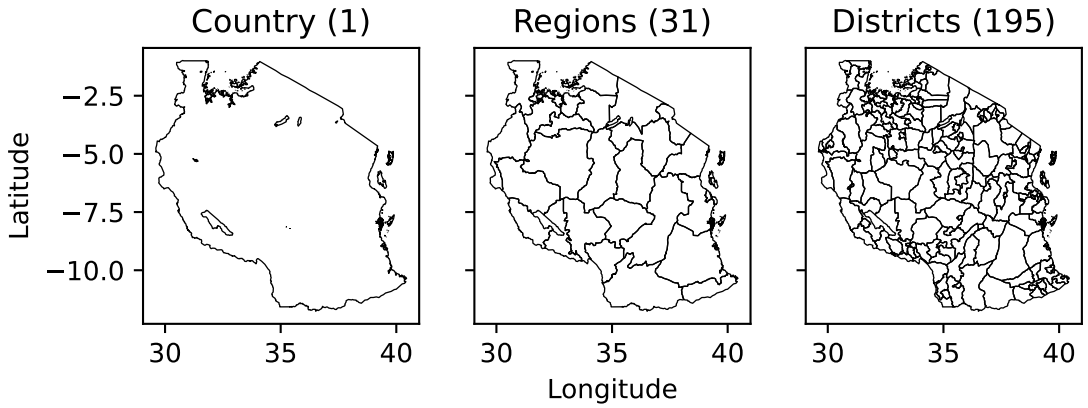
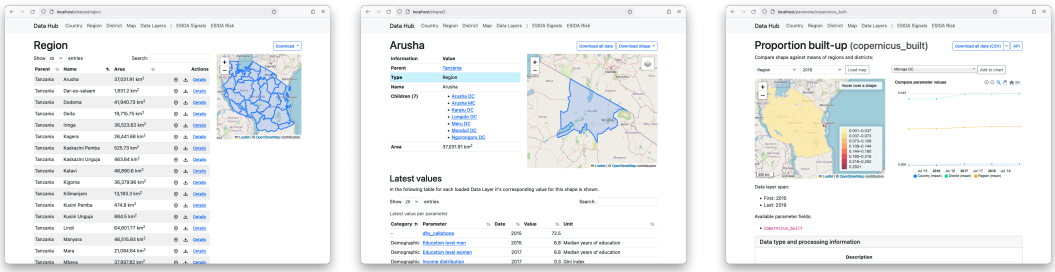


Abbildung 4.3: Im Data Hub importiere Shapes von Tansania in ihren jeweiligen Kategorien.

Über das Webinterface des Data Hubs lassen sich die gesammelten Daten durchsuchen und erste Bewertungen durchführen (siehe Abb. 4.4). Zusätzlich sind über diesen Weg die Zugriffsmöglichkeiten mittels API auf den jeweiligen Seiten von Shapes oder Parameters dokumentiert. Außerdem ermöglicht das Webinterface die Exploration der integrierten Data Layer und dient damit als Bindeglied innerhalb des Projektes. Da es sich *nur* um eine Website handelt, ist der Zugriff ohne technische Hürden oder benötigte technische Kompetenzen möglich.

[3] https://www.nbs.go.tz/index.php/en/census-surveys/gis/568-tanzania-districts-shapefiles-2019



(a) Übersicht Shapes **(b)** Detailansicht Shape **(c)** Ansicht eines Data Layers

Abbildung 4.4: Verschiedene Ansichten des Webinterfaces zum Einsehen und Bewerten der aggregierten Datenquellen. Großansicht in Abb. A.2 bis A.4.

4.3 Risiko-Algorithmus Tansania

Im Kontext des ESIDA-Projektes wurde für Tansania und die relevanten Faktoren von Denguefieber ein Risiko-Algorithmus definiert. Dieser basiert auf der Einordnung verschiedener Parameter in drei verschiedene Grenzwerte. Je nach Einordnung wird ein Score von eins, zwei oder drei vergeben. Die Summe über alle Parameter hinweg liefert den Risiko-Score. Der Risiko-Score wird auf zwei Achsen definiert, einmal der Wahrscheinlichkeit, dass Denguefieber vorliegt (*Likelihood*[4]) und dem möglichen Risiko eines größeren Ausbruchs (*Impact*[5]), siehe Abschnitt 2.1. Die Grenzwerte sind durch Mitglieder des ESIDA-Projektes definiert worden. Beide Algorithmen wurden mittels der zuvor vorgestellten Workflow-Engine ausgeführt und die Ergebnisse in zwei Data Layern gespeichert (`esida_risk_impact` und `esida_risk_likelihood`). Für den Impact ist in Abb. 4.5 exemplarisch der Durchlauf und das Ergebnis für die Region Daressalam am 1.1.2020 dargestellt.

[4] https://github.com/MARS-Group-HAW/esida-db/blob/main/input/algorithms/esida-risk-assessment-likelihood.yaml
[5] https://github.com/MARS-Group-HAW/esida-db/blob/main/input/algorithms/esida-risk-assessment-impact.yaml

	Risk Factor	Risk Score			Example: *Dar-es Salaam (2020-01-01)*		
		1	*2*	*3*	Value	Score	Sum
Health care capacity and access	Physician density (per 10,000 population) tnbs_physdens	> 29	8 - 29	< 8	5,18	3	3
	Nurses and midwife density (per 10,000 population) tnbs_nursmidwdens	> 71	17 - 71	< 17	3,1	3	6
	Hospital beds per 10,000 population who_hospbeddens	> 40	20 - 24	< 20	7	3	9
	Healthcare accessibility (optimal motorized travel time to nearest facility) malariaatlas_traveltimehc	< 8	8 - 15	> 15	2,879	1	10
Socio-demographic	Proportion of population aged 5-39 years worldpop_age_5_39/worldpop_popc	< 50,3	50,3 - 62,8	> 62,8	67,85	3	13
	Proportion of population aged < 15 years worldpop_age_0_14/worldpop_popc	< 20,4	20,4 - 30,9	> 30,9	29,23	2	15
	Proportion of population aged + 65 years worldpop_age_65/worldpop_popc	< 4,9	4,9 - 10,9	> 10,9	1,71	1	16
						Risk-Score:	**16**

Abbildung 4.5: Risk-Score-Algorithmus für Impact, mit Beispiel Daressalam.

Da für jeden abgefragten Parameter innerhalb der Algorithmen jeweils ein Wert zwischen eins und drei vergeben wird, liegt der mögliche Wertebereich des Impacts bedingt durch sieben Parameter im Bereich zwischen 7 und 21. Im Falle der Likelihood handelt es sich um 23 Parameter, womit der Wertebereich zwischen 23 und 69 liegt. Ein höherer Wert entspricht jeweils einem höheren Risiko. Da die Workflow-Engine den Risiko-Score über die Zeit mit den jeweils gültigen Parametern erstellt, kann der Risiko-Score je Shape im zeitlichen Verlauf analysiert werden, wie in Abb. 4.6 zu sehen. Dabei ist zu sehen, dass der Wert des Likelihood-Scores nur geringfügig variiert. Gut zu erkennen ist der Effekt der Saisonalität, die je nach entsprechendem Wetter, das Risiko erhöht oder reduziert. Im Fall des Impacts verändert sich der Wert kaum. Lediglich ab 2014 reduziert sich der Score um einen Punkt. Dies geht auf eine Abnahme der Bevölkerungsanteile jünger als 15 Jahre zurück (Data Layer `worldpop_age_0_14`), wodurch ein geringerer Score vergeben wird. Neben der temporalen Übersicht ist auch der räumliche Vergleich relevant, um regionale Unterschiede zu erkennen. In Abb. 4.7 ist dabei der jeweilige Score von Likelihood und Impact als Choroplethenkarte dargestellt.

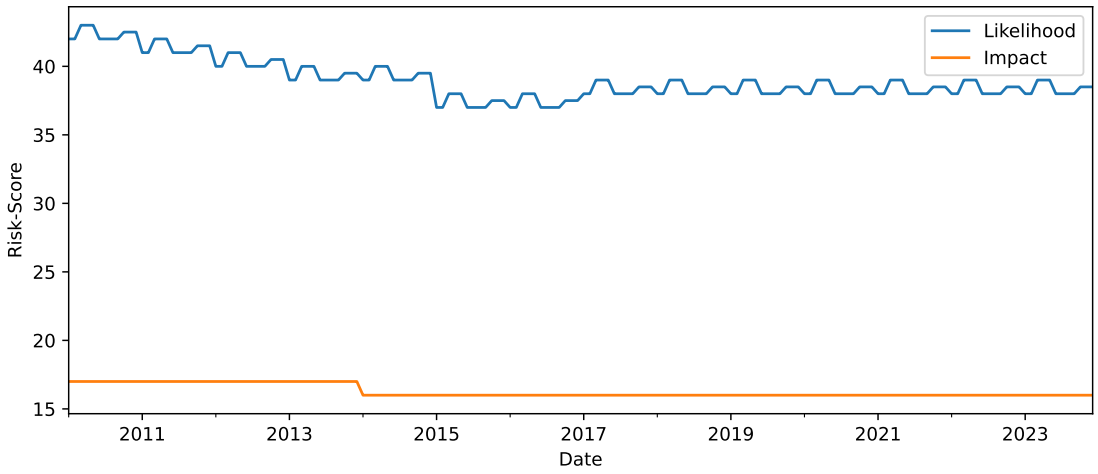
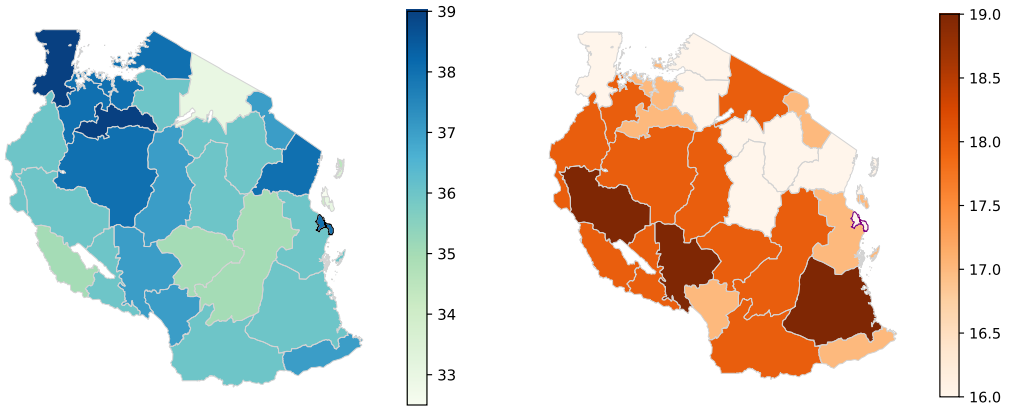


Abbildung 4.6: Der Risiko-Score von Likelihood und Impact im zeitlichen Verlauf, für die Region Daressalam.



(a) Likelihood **(b)** Impact

Abbildung 4.7: Landesweiter Vergleich des Risikos zum Zeitpunkt des 1.1.2020.

Um das kombinierte Risiko für die Region einschätzen zu können, bietet sich die Darstellung der beiden Scores auf zwei Achsen an. In Abb. 4.8 ist dies als Konturdiagramm geschehen, was Scatterplot und Heatmap vereint. Die hier exemplarisch betrachtete Region Daressalam ist als schwarzer Punkt bei $(16, 38)$ aufgetragen. Dies ermöglicht die schnelle Einschätzung einer spezifischen Region im Kontext Tansanias. Die Einfärbung der Konturen stellt die Verteilung aller Regionen in Tansania dar.

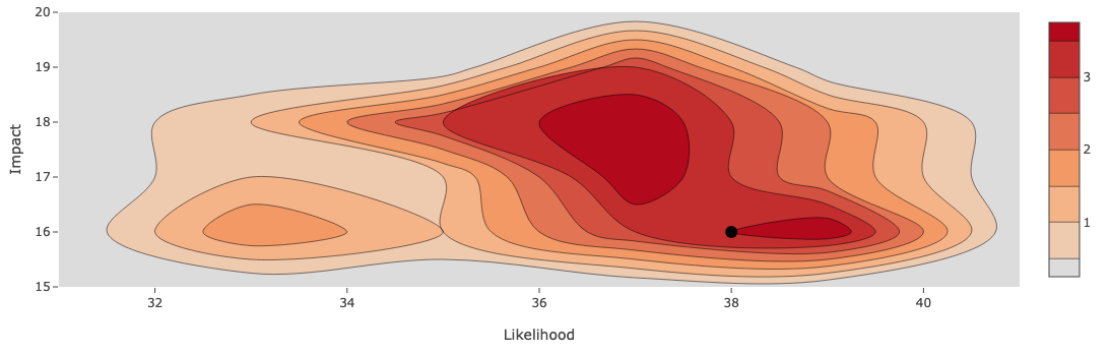


Abbildung 4.8: Der Risiko-Score von Likelihood und Impact zum Zeitpunkt des 1.1.2020 als Konturdiagramm, das die Verteilung aller Regionen angibt. Die Region Daressalam ist als schwarzer Punkt verortet.

Die Abb. 4.6 bis 4.8 befinden sich analog im Data Hub Webinterface auf der spezialisierte Unterseite zum Risk-Score[6]. Für jede AoI (District, Region, Country) lassen sich die Diagramme jeweils für jeden Zeitpunkt dynamisch erstellen. Damit kann ein Anwender historische und aktuelle Lagen beurteilen. Um die Transparenz des Systems zu steigern, steht zusätzlich für jeden Zeitpunkt ein Log zur Verfügung, an dem genau abgelesen werden kann, welcher Parameter mit welchem Wert in die Ermittlung der jeweiligen Scores eingeflossen ist, wie exemplarisch für Likelihood in Abb. 4.9 zu sehen. Diese Systematik ist nicht an den ESIDA Risiko-Score gekoppelt, sondern steht für jeden definierten Algorithmus Data Layer zur Verfügung.

[6]Menüpunkt *ESIDA Risk* im Webinterface des Data Hubs.

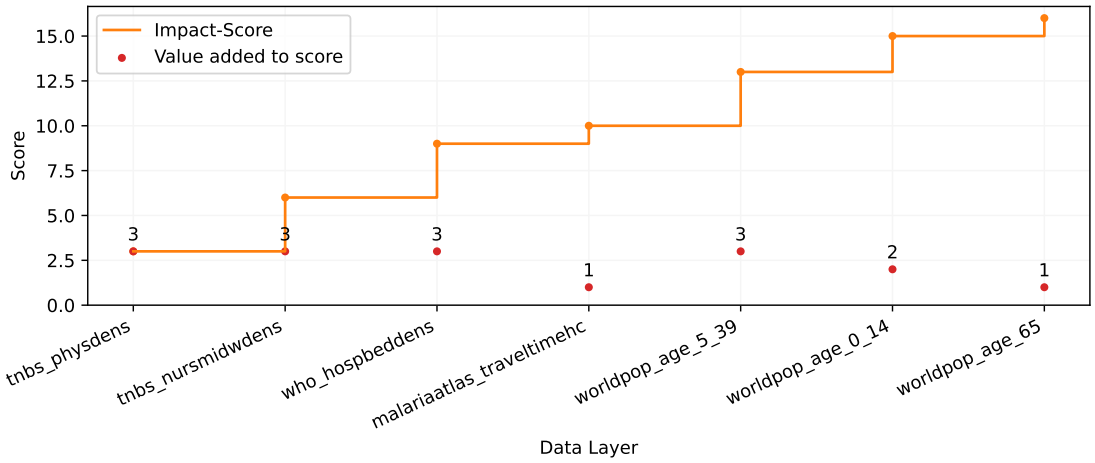


Abbildung 4.9: Visualisierung des Logs der Zusammensetzung des Likelihoods-Scores für den 1.1.2020 und die Region Daressalam.

4.4 Signal-Bewertung

Nach der manuellen Eingabe eines Falls über das Webinterface wird automatisch die Einschätzung dynamisch erstellt, wie in Abb. 4.10 zu sehen. Hierbei ist anzumerken, dass im Rahmen des Projektverlaufes die ursprünglich angedachte automatische Integration von Signalen in das System nicht realisiert wurde. Damit zusammenhängend wurde der Algorithmus für die Ad-hoc-Bewertung nicht vollständig entwickelt. Dennoch ist die Systematik so weit implementiert, dass exemplarisch die Funktionsweise demonstriert werden kann. Der prototypische Algorithmus fragt dabei zuerst die minimale Temperatur in den 14 Tagen vor der Meldung ab. Sollte sich hierbei bereits zeigen, dass der Grenzwert von 15 °C unterschritten wurde, würde der Algorithmus abbrechen. Da das Dengue-Virus bei solchen Temperaturen nicht mehr überlebensfähig ist, ist es sehr unwahrscheinlich, dass es sich um eine Infektion mit Dengue handelt. In der in Abb. 4.10 dargestellten Situation, ist dies aber nicht der Fall. Zusätzlich ist der Urbanisierungsgrad größer als 10 %, wodurch eine schnelle Ausbreitung wahrscheinlich wäre, was zur Einschätzung eines hohen Risikos für einen Ausbruch von Dengue führt.

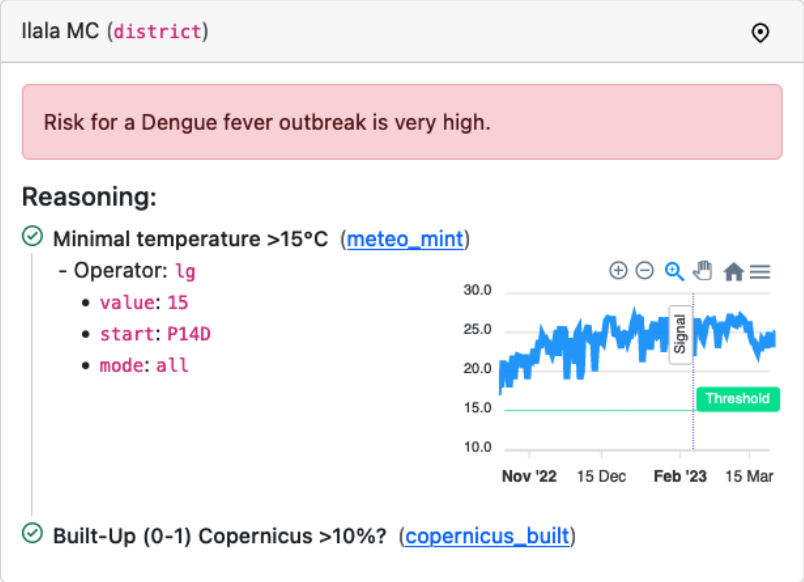


Abbildung 4.10: Ad-hoc-Bewertung eines Signals im Ilala Distrikt (Gesamtseite, siehe Abb. A.1).

4.5 Performance

Die Geschwindigkeit des Gesamtsystems lässt sich nur schwer beurteilen, da es hierbei zu vielen unbekannten Größen kommt. Bei der Integration der Daten hängt dies u. a. von dem Netzwerk zwischen Quellsystem und Data Hub ab. Die Dauer der Verarbeitung der Daten unterliegt dem Umfang der Daten. Allerdings lassen sich äquivalente Operationen in einem anderen GIS mit dem Data Hub vergleichen. Dazu wurde ein weltweiter Raster-Datensatz[7] herangezogen, in dem die durchschnittliche benötigte Zeit bis zur nächsten Gesundheitseinrichtung in einem $1 \times 1\,\text{km}$ Raster modelliert wurde. Dieser Datensatz wurde mit Grenzen aller Länder weltweit[8] verschnitten und für jedes Land die durchschnittliche Zeit zum Erreichen einer Gesundheitseinrichtung berechnet. Diese Operation wurde jeweils im Data Hub ausgeführt und in QGIS (v3.30.3) mit dem „Zonal statistics“ Plugin.

[7] https://doi.org/10.1186/s12936-018-2500-5
[8] https://datacatalog.worldbank.org/search/dataset/0038272/World-Bank-Official-Boundaries, World Boundaries GeoJSON - Low Resolution.

Die Operation wurde jeweils 10 Mal auf demselben Computer[9] ausgeführt und die Ausführungszeit gemittelt. Im Data Hub beträgt die Zeit 20,5 s, bei QGIS 262,5 s (4:22 Minuten). Innerhalb von QGIS streut die Dauer sehr, wie in Abb. 4.11 zu sehen ist, wohingegen sie im Data Hub nur einer sehr geringen Abweichung unterliegt. Die Operation ist im Data Hub um mehr als den Faktor 10 schneller. Der Unterschied lässt sich mit den umfangreichen Optimierungen für die Verarbeitungen von Matrizen erklären, die in den vom Data Hub verwendeten Bibliotheken (NumPy, Rasterio) vorliegen. Es wurden im Rahmen dieser Arbeit aber keine weiteren Untersuchungen dahingehend angestellt. Dennoch zeigt dies, dass deutliche Geschwindigkeitsgewinne zu erreichen sind, wenn spezialisierte Werkzeuge entwickelt werden.

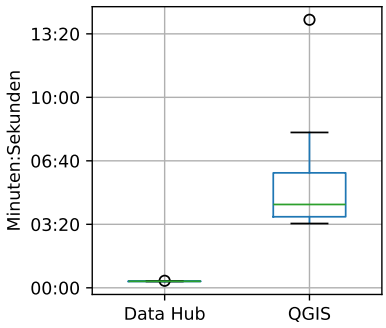


Abbildung 4.11:
Ausführungsdauer der Ermittlung der Mittelwerte eines Rasters für Polygone.

[9] MacBook Pro (14”, 2021), M1 Max, 32 GB RAM

5 Diskussion

Die Realisierbarkeit des Systems kann mit den gewonnenen Ergebnissen bestätigt werden. Durch die flexible Architektur ist es möglich, eine große Zahl verschiedener Daten zu integrieren. Die Architektur beruht dabei auf unterschiedlichen Herangehensweisen. Sie beinhaltet Aspekte eines Data Lakes, da die ursprünglichen Daten in ihrer Originalfassung vorgehalten werden. Weiter beinhaltet sie aber auch Aspekte des Data Warehouses, da aggregierte Daten auf zeitlicher und räumlicher Dimension bereitgestellt werden. Die Darstellung der Daten anhand mehrerer Dimensionen entspricht weiterhin dem Konzept eines Data Cubes. Zusätzlich greift das System bestehende Konzepte anderer Werkzeuge auf und enthält die Metadaten der jeweiligen Rohdaten, was einem Data Repository entspricht. Mit der umfangreichen Workflow-Engine erlaubt es zudem automatisierte und reproduzierbare Arbeitsschritte eines klassischen GIS. In der Synthese ergibt sich daraus der Name des Data Hubs.

Mit dem zusätzlich entwickelten Webinterface ist eine Zugriffsmöglichkeit gegeben, die nicht an einen einzelnen Computer oder eine Plattform gebunden ist. Damit kann es auch in größeren Teams als zentrale Plattform genutzt werden, die eine gemeinsame Datenbasis ermöglicht. Zudem wird es damit auch technisch nicht versierten Akteuren erlaubt, mit umfangreichen Datenmengen zu arbeiten und diese, durch die Downloadfunktionen oder API, in jeweils vertrauerten Umgebungen auszuwerten. Darüber hinaus bieten die Auswertungen und Visualisierungen innerhalb des Data Hubs bereits einen schnellen Überblick über die Daten, was von Epidemiologen genutzt werden kann, um sich schnell ein Lagebild zu verschaffen. Der Data Hub bildet damit ein umfangreiches Entscheidungsunterstützungssystem.

5.1 Open Source

Die Bereitstellung des Data Hubs unter der MIT-Lizenz und damit als Open Source ermöglicht zudem jedem Interessierten, das System zu benutzen und zu verstehen. Somit

sind die ermittelten Ergebnisse nachvollziehbar und transparent. Dies ist vor allem in Gesundheitsanwendungen von großer Wichtigkeit, da Fehler oder Einschränkungen des Systems bekannt sein müssen, um die abgeleiteten Empfehlungen einordnen zu können. Darüber hinaus erlaubt es, dass das System an individuelle Bedarfe durch den Anwender selbst angepasst werden kann und somit optimal in eine bestehende Infrastruktur integriert werden kann.

Zwar besteht das Risiko, dass sich das System in vielerlei Richtungen entwickelt, wenn es zu parallelen Entwicklungen oder Abspaltungen kommt. Dennoch können durch ein entsprechend transparentes Community-Management eine Vielzahl von Synergien gewonnen und doppelte Entwicklungen vermieden werden. Die Ausgestaltung als schlankes Framework mit den Möglichkeiten einer Plugin-Struktur kann das Gesamtsystem stärken. Außerdem ermöglicht es den Einsatz in völlig neuen Anwendungsgebieten, die durch die Entwickler nie vorgesehen waren. Dies kann neue Ansätze und generelle Verbesserungen in das Projekt einbringen.

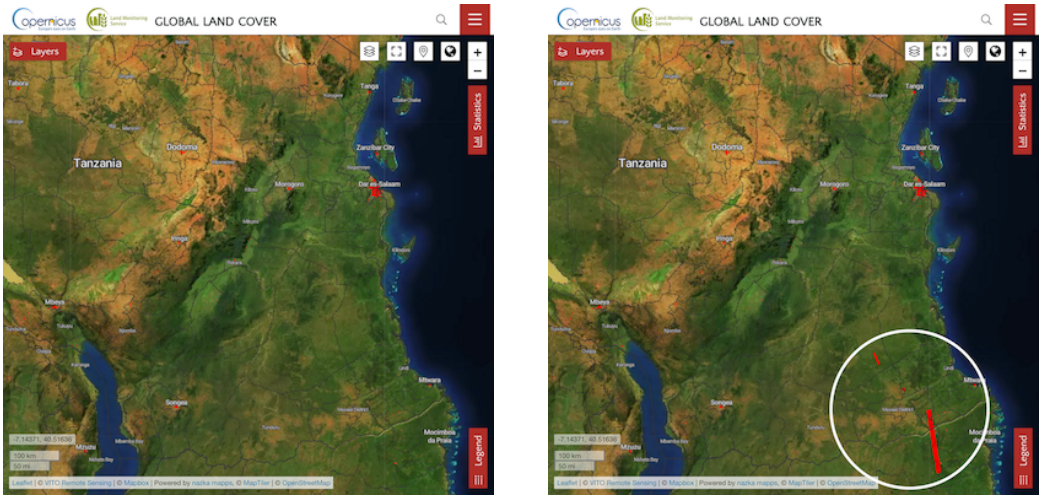
Des Weiteren erlaubt das Anbieten der Software als Open Source den Einsatz in Gebieten, wo keine finanziellen Möglichkeiten zum Kauf von Software bestehen. Darüber hinaus können Erfahrungen und Best-Practices kommuniziert werden, was gerade in den Ländern des globalen Südens zu einer Unterstützung führen kann (Capacity building).

5.2 Datenqualität

Die Qualität der Daten ist immer von den spezifischen Anforderungen des Projektes abhängig. Im Data Hub kann daher keine allgemeingültige Einschätzung der Datenqualität gegeben werden. Der Data Hub versteht sich als Plattform und nicht als spezifisches Werkzeug zur Beurteilung der Qualität. Die Qualität der Daten ist immer durch den Anwender selbst zu bestimmen und damit obliegt es ihm selbst, passende Daten zu integrieren und vorab zu bestimmen, ob sie sich für den Anwendungsfall eignen.

Dennoch kann durch die Analysemöglichkeiten im Data Hub ein zusätzlicher Eindruck über die Daten gewonnen werden. Mit den Heatmaps und Zeittrends in den verschiedenen Ansichten des Data Hubs liefert er einen schnellen Überblick über die räumliche und zeitliche Verteilung der Daten. Durch die vorhandenen Metadaten lassen sich außerdem weitere Eigenschaften der Daten schnell bestimmen oder die Rohdaten direkt aufrufen. Dies ist vor allem hilfreich bei der Analyse von Ausreißern. Konkret ließ sich so erklären,

wieso in den Copernicus Landnutzungsdaten im tansanischen Distrikt Newala TC der Anteil der bebauten Fläche von 3,2 % im Jahr 2016 auf 37,1 % im Jahr 2017 angestiegen ist. Hierbei handelte es sich um einen offensichtlichen Fehler in den Rohdaten, wie in dem in Abb. 5.1b eingekreisten Bereich deutlich zu erkennen, im Kontrast zu Abb. 5.1a.



(a) 2016 **(b)** 2017

Abbildung 5.1: Vergleich Copernicus Landnutzung, bebaute Landfläche in rot. Screenshot von https://lcviewer.vito.be/.

Für tiefergehende Analysen der integrierten Daten stehen dem Anwender die API oder Downloads zur Verfügung. Damit können umfangreiche Auswertungen erstellt werden, die verschiedene Data Layer mit einbeziehen. Da der Anwender diese Untersuchungen direkt in der ihm bekannten Umgebung durchführen kann, wie etwa R oder Python, führt dies zu einer höheren Arbeitsgeschwindigkeit des Users, da keine Einarbeitung stattfinden muss.

5.3 Reproduzierbarkeit

Die Reproduzierbarkeit ist eines der Hauptmerkmale des Data Hubs. Damit können, wie dargestellt, Ergebnisse jederzeit erneut hergestellt werden, bzw. sollte es zu neuen Daten kommen oder Änderungen bei den Bewertungsalgorithmen geben, die Assessments neu ermittelt werden. Dabei ist kein weiterer manueller Aufwand erforderlich (außer die Integration der Änderungen). Dies erlaubt zusätzlich das schnelle Testen von Variationen

oder Alternativen und deren Vergleich. Voraussetzung dafür ist allerdings, dass die Rohdaten unverändert sind und die entsprechenden Quellen die Daten in der gleichen Form bereitstellen wie zum Erstellungszeitpunkt. Bereits während der Entwicklung gab es Änderungen bei einzelnen Datenquellen, wie bspw. dem Malariaatlas. In diesem Fall wurde die Architektur der Website, welche die Daten bereitstellt, geändert. Hierzu musste die Integration einmalig angepasst werden, anschließend konnten aber alle Funktionen im Data Hub direkt wieder ausgeführt werden. Das Risiko von solchen Änderungen besteht jederzeit. Hierbei könnten nur formalisierte und weiträumig akzeptierte Forschungsdatenstrukturen Abhilfe schaffen, die Forschenden langfristige Speicherung von Daten garantieren.

Dennoch bietet der Data Hub hier eine Lösung, da die Daten immer physisch integriert werden, und somit die Rohdaten nach dem Zeitpunkt der Integration unverändert vorliegen. Dies reduziert zum einen die Last bei den Quellsystemen, da die Daten nicht bei jeder Änderung einer Verarbeitung neu abgerufen werden, zum anderen ermöglicht es alle Rohdaten eines Projektes gesammelt zu archivieren und somit auch zukünftig die Ergebnisse wieder herstellen zu können, sollte der Fall eintreten, dass eine Datenquelle ihre Daten nicht mehr bereitstellt.

5.4 Transfer in Epidemiologie und Public Health

Der Data Hub ist im Rahmen des Projektes zu einem wertvollen Werkzeug geworden, das für die Erstellung von Lagebildern und Einschätzung von Risiken zu akuten Situationen verwendet werden kann. Neben den Anwendungsgebieten in der Epidemiologie zur Bewertung von Krankheitsausbrüchen ist auch der Einsatz in weiteren Public-Health-Szenarien denkbar, wie bspw. in der Planung und Ausführung von Hilfseinsätzen bei Naturkatastrophen, indem die Kontextdaten wertvolle Ergänzungen liefern können.

Zusätzlich dient die Übersicht der integrierten Daten in Forschungsteams als gemeinsame Basis, und kann damit doppelte Aufwände bei der Recherche vermeiden. Forschende können durch die mit Metadaten kuratierten Datensätze und die Verlinkung zu den Rohdaten auf den Data Hub zurückgreifen. Bei der Entwicklung von Forschungsfragen kann so auf die bereits erfolgten Recherchen vergangener Projekte zurückgegriffen werden. Zusätzlich bietet der Data Hub mit den Visualisierungen einen schnellen Überblick zu den Ergebnissen vorangegangener Fragestellungen.

Neben den Anforderungen an den Data Hub aus dem ESIDA-Projekt, konnte das System auch in weiteren Szenarien angewendet werden. Im Rahmen einer Lehrveranstaltung in der HAW Hamburg am Department Gesundheitswissenschaften und bei der EARTHS Summerschool[1] wurde der Data Hub im Themenfeld Epidemic Intelligence eingesetzt. In beiden Fällen wurden den Studierenden Aufgabenstellungen aus dem Bereich Public Health gegeben, für deren Beantwortung unterschiedliche Daten benötigt wurden, wie bspw. Bevölkerungszahlen oder die Verfügbarkeit von Gesundheitseinrichtungen. In diesen Anwendungsbeispielen fungierte der Data Hub als Datenrepository, in dem die Studierenden eigenständig nach den entsprechend benötigten Daten suchen konnten. Die Präsentation der aufbereiteten Daten auf räumlicher und zeitlicher Ebene unterstützte dabei die Studierenden zusätzlich. Dieses Szenario unterstreicht die Synergien, die aus der interdisziplinären Forschung entstehen können, in diesem Fall zwischen Informatik und Public Health.

5.5 Weitere Anwendungsszenarien

Neben dem in dieser Arbeit vorgestelltem Anwendungsfall eines EWARS stellt der Data Hub aber auch ein flexibles GIS dar, welches in anderen Domänen Anwendung finden kann. Ein Beispiel für diese Anwendung wurde bereits in Abschnitt 4.5 angedeutet. Mit einfachen Mitteln können durch den Data Hub reproduzierbare Auswertungen auf beliebigen Raster- und Vektordaten ausgeführt werden. Dies wird durch die generische Grundstruktur erreicht, die es erlaubt, beliebige Daten zu integrieren.

Mittels der vorgestellten Workflow-Engine lassen sich neben einem Risiko-Assessement auch andere Fragestellungen beantworten. Darunter fallen sämtliche Szenarien, in denen Geodaten mit beliebigen weiteren Daten verschnitten werden und ein anschließendes Ranking vorgenommen werden muss. Hypothetische Szenarien könnten zum Beispiel die Planung der Zuordnung von Hilfsgütern in Krisengebieten anhand von Bevölkerungsdichte und Straßeninfrastruktur sein oder die Planung von Reisen anhand gewünschter PoIs im Zielland und der gemittelten historischen Temperatur.

Neben diesen legitimen Anwendungsfällen, stellt sich allerdings auch die Frage, ob die Software in weiteren Szenarien eingesetzt werden könnte, die nicht im Sinne des Allgemeinwohls stehen. Darunter könnte eine militärische oder gar terroristische Nutzung

[1] https://www.haw-hamburg.de/en/university/faculty-of-life-sciences/departments/health-sciences/earths/earths-2023/

fallen, wie etwa das algorithmische Ermitteln von Angriffszielen mit dem größtmöglichen Schaden. Die Potenziale einer solchen Nutzung müssten in weiteren Studien untersucht und abgewogen werden. Klar steht aber der Mehrwert und der Nutzen von Open Source im Vordergrund und damit eben die freie Verfügbarkeit. Besonders durch die Identifikation als Open Source, entfällt die Diskriminierung von Einsatzgebieten der Software (siehe Abschnitt 2.3), aber damit auch der explizite Ausschluss einer militärischen Nutzung.

6 Zusammenfassung

Zusammengefasst ist mit dem Data Hub ein flexibles und vielseitiges Geoinformationssystem entstanden. Die Bedarfe des ESIDA-Projektes gaben dabei die Anforderungen aus der Domäne der Epidemiologie bzw. der Gesundheitswissenschaften vor. Durch den intensiven interdisziplinären Austausch konnten sich diese mit den Bestrebungen zur Generalisierung aus der Informatik und den eingeflossenen Best Practices zu dem nun vorliegenden *Data Hub* entwickeln. Nur durch die Synthese dieser unterschiedlichen Sichtweisen konnte das System in dieser Fassung letztlich entwickelt werden, was veranschaulicht, wie wichtig agiles und interdisziplinäres Arbeiten ist.

Mit dem so entwickelten Werkzeug lassen sich umfangreiche Datenrepositorien erstellen, die als zentrales Bindeglied für den Datenaustausch in einem diversen Team dienen können und damit eine gemeinsame Datenbasis etablieren. Zudem gestaltet sich der Zugriff mit dem Webinterface, nachdem das System einmal installiert wurde, ohne große Hürden. Zusätzlich erlaubt die, auf den Daten basierende, Workflow-Engine das Abfragen der verschiedenen Daten in einer automatisierten Form. Dies wird im Rahmen des ESIDA-Projektes zur Erstellung einer Risiko-Matrix für Denguefieber in Tansania verwendet, sowie zur Ad-hoc-Beurteilung von Krankheitsmeldungen. Damit leistet das System einen wichtigen Beitrag zum One-Health-Ansatz, der in Zeiten von weltweit strapazierten Gesundheitssystemen immer wichtiger wird.

Weiterhin ist das Gesamtsystem vollständig reproduzierbar, was – bedingt durch den Cyberangriff auf die HAW Hamburg – bereits in einem realen Szenario erfolgreich erprobt wurde. Dadurch, dass die Datenintegration, die Verarbeitung und die Auswertung durch den Quellcode definiert sind, können sie immer neu ausgeführt werden (solange die Rohdaten vorliegen). Die Reproduzierbarkeit von Ergebnissen durch Dritte ist eine der Säulen des wissenschaftlichen Arbeitens. Der Data Hub ermöglicht dies durch die Vermeidung manueller, undokumentierter Schritte, sowie durch die Veröffentlichung unter der MIT Open-Source-Lizenz.

Der Python-Quellcode des Systems ist offen auf GitHub[1] einsehbar und damit durch jeden überprüfbar, was gerade für ein Entscheidungsunterstützungssystem ein zentraler Aspekt ist. Damit können die gewonnenen Empfehlungen auf Ebene des Quellcodes jederzeit nachvollzogen werden. Zusätzlich erlaubt die offene Lizenzierung die Erweiterung und Anpassung an lokale Gegebenheit und verhindert Unsicherheiten in Hinblick auf Kosten oder erlaubte Nutzung. Dies ist besonders im Gebiet der Krisenintervention relevant, wo schnell Entscheidungen getroffen werden müssen.

Die Verwendung von Python und der Einsatz des Flask-Microframeworks hat sich für die Entwicklung bewährt. Zum einen konnte auf das umfangreiche Ökosystem von Python zurückgegriffen werden, was im Bereich der Geodatenverarbeitung stark ausgeprägt ist, zum anderen erlaubt Flask eine schnelle Entwicklung und Prototyping und damit gerade im Bereich des agilen Arbeitens schnelle Iterationsschleifen und Ergebnisse, die im Team diskutiert werden konnten.

6.1 Ausblick

Bei der Weiterentwicklung des Projektes gibt es u. a. die folgenden Aspekte, die bisher nicht, oder nur zum Teil, berücksichtigt wurden: Allem voran steht dabei, eine Update-Methodik für den Quellcode zu integrieren. Dies war im Kontext des Projektes kein Kriterium, da es sich um ein einzelnes Projekt handelte. Es hat sich aber bereits in der Entwicklung gezeigt, dass durch unterschiedliche Instanzen des Data Hubs, mit jeweils unterschiedlich ausgeprägten Data Layern, eine Methodik zur Synchronisierung des Applikationscodes hilfreich gewesen wäre. Dies könnte bspw. über ein Python-Package erfolgen, das mittels der pip-Paketverwaltung verteilt wird. Zusätzlich hat sich die Notwendigkeit einer Verwaltung von Benutzern und deren Rechten erwiesen. Da aktuell lediglich ein globaler Zugriffsschutz mittels eines vorgeschalteten Proxys möglich ist, gibt es keine Möglichkeit, befristete Zugänge einzurichten oder granular einzelne Data Layer öffentlich oder privat zu schalten. Dies würde sich als hilfreich bei der Bereitstellung vorläufiger Ergebnisse ausschließlich innerhalb des Projektteams erweisen. Die Performance des Systems stand während der Entwicklung nicht im Fokus. Da es sich aber zu großen Teilen um Verarbeitungsprozesse ohne Abhängigkeiten handelt, sind sie „embarrassingly parallel“ [35]. Durch Parallelisierung der Verarbeitung ließen sich folglich große Beschleunigungen erzielen.

[1] https://github.com/MARS-Group-HAW/esida-db

Neben der technischen Weiterentwicklung muss vor allem eine weitere Validierung der Ergebnisse in Hinblick auf die Epidemiologie und damit die Funktion als EWARS erfolgen. Dafür werden in weiteren interdisziplinären Teams neue Forschungsgebiete und Algorithmen für andere Szenarien erarbeitet.

Literatur

[1] Nicole L. Achee et al. „A Critical Assessment of Vector Control for Dengue Prevention“. In: *PLOS Neglected Tropical Diseases* 9.5 (7. Mai 2015), e0003655. ISSN: 1935-2735. DOI: 10.1371/journal.pntd.0003655.

[2] Rokeya Akter et al. „Joint Effects of Climate Variability and Socioecological Factors on Dengue Transmission: Epidemiological Evidence“. In: *Tropical Medicine & International Health* 22.6 (2017), S. 656–669. ISSN: 1365-3156. DOI: 10.1111/tmi.12868.

[3] Allianz der deutschen Wissenschaftsorganisationen. *Grundsätze zum Umgang mit Forschungsdaten*. 2010. DOI: 10.2312/ALLIANZOA.019.

[4] Ayansina Ayanlade et al. „Early Warning Climate Indices for Malaria and Meningitis in Tropical Ecological Zones“. In: *Scientific Reports* 10.1 (1 31. Aug. 2020), S. 1–13. ISSN: 2045-2322. DOI: 10.1038/s41598-020-71094-8.

[5] Ben Balter. *Open Source License Usage on GitHub.Com*. The GitHub Blog. 10. März 2015. URL: https://github.blog/2015-03-09-open-source-license-usage-on-github-com/ (besucht am 07. 08. 2023).

[6] Samir Bhatt et al. „The Global Distribution and Burden of Dengue“. In: *Nature* 496.7446 (7446 Apr. 2013), S. 504–507. ISSN: 1476-4687. DOI: 10.1038/nature12060.

[7] Filip Biljecki, Yoong Shin Chow und Kay Lee. „Quality of Crowdsourced Geospatial Building Information: A Global Assessment of OpenStreetMap Attributes“. In: *Building and Environment* 237 (1. Juni 2023), S. 110295. ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2023.110295.

[8] David E. Bloom und Daniel Cadarette. „Infectious Disease Threats in the Twenty-First Century: Strengthening the Global Response“. In: *Frontiers in Immunology* 10 (2019). ISSN: 1664-3224.

[9] BMWi. *Das Projekt GAIA-X*. Okt. 2019, S. 51.

[10] Geoff Boeing. „OSMnx: New Methods for Acquiring, Constructing, Analyzing, and Visualizing Complex Street Networks“. In: *Computers, Environment and Urban Systems* 65 (2017), S. 126–139. ISSN: 0198-9715. DOI: 10.1016/j.compenvurbsys.2017.05.004.

[11] J Boenecke et al. „Harnessing the Potential of Digital Data for Infectious Disease Surveillance in Sub-Saharan Africa“. In: *European Journal of Public Health* 32 (Supplement_3 1. Okt. 2022), ckac131.569. ISSN: 1101-1262. DOI: 10.1093/eurpub/ckac131.569.

[12] Will Brackenbury et al. „Draining the Data Swamp: A Similarity-based Approach“. In: *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*. HILDA '18. New York, NY, USA: Association for Computing Machinery, 10. Juni 2018, S. 1–7. ISBN: 978-1-4503-5827-9. DOI: 10.1145/3209900.3209911.

[13] Jobie Budd et al. „Digital Technologies in the Public-Health Response to COVID-19“. In: *Nature Medicine* 26.8 (8 Aug. 2020), S. 1183–1192. ISSN: 1546-170X. DOI: 10.1038/s41591-020-1011-4.

[14] Leonardo Candela et al. „Data Journals: A Survey“. In: *Journal of the Association for Information Science and Technology* 66.9 (2015), S. 1747–1762. ISSN: 2330-1643. DOI: 10.1002/asi.23358.

[15] CDC. *Health Alert Network (HAN) - 00494 | Locally Acquired Malaria Cases Identified in the United States*. 26. Juni 2023. URL: https://emergency.cdc.gov/han/2023/han00494.asp (besucht am 16. 08. 2023).

[16] Emily H. Chan et al. „Global Capacity for Emerging Infectious Disease Detection“. In: *Proceedings of the National Academy of Sciences* 107.50 (14. Dez. 2010), S. 21701–21706. DOI: 10.1073/pnas.1006219107.

[17] Kang-tsung Chang. *Introduction to Geographic Information Systems*. 9. Aufl. 30. Jan. 2018. ISBN: 978-1-259-92964-9.

[18] *COMMUNICATION FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT, THE COUNCIL, THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE REGIONS European Cloud Initiative - Building a Competitive Data and Knowledge Economy in Europe*. 2016.

[19] Paul Currion, Chamindra de Silva und Bartel Van de Walle. „Open Source Software for Disaster Management“. In: *Communications of the ACM* 50.3 (1. März 2007), S. 61–65. ISSN: 0001-0782. DOI: 10.1145/1226736.1226768.

[20] Alexandre Decan und Tom Mens. „What Do Package Dependencies Tell Us about Semantic Versioning?“ In: *IEEE Transactions on Software Engineering* 47.6 (2021), S. 1226–1240. DOI: 10.1109/TSE.2019.2918315.

[21] ECDC. *Increasing Risk of Mosquito-Borne Diseases in EU/EEA Following Spread of Aedes Species*. URL: https://www.ecdc.europa.eu/en/news-events/increasing-risk-mosquito-borne-diseases-eueea-following-spread-aedes-species (besucht am 16. 08. 2023).

[22] Lisa Ehrlinger et al. „Data Catalogs: A Systematic Literature Review and Guidelines to Implementation“. In: *Database and Expert Systems Applications - DEXA 2021 Workshops*. Hrsg. von Gabriele Kotsis et al. Cham: Springer International Publishing, 2021, S. 148–158. ISBN: 978-3-030-87101-7.

[23] European Centre for Disease Prevention and Control. *Operational Tool on Rapid Risk Assessment Methodology: ECDC 2019*. ECDC Technical Report. LU: European Centre for Disease Prevention and Control, 2019. ISBN: 978-92-9498-309-1.

[24] Anthony S. Fauci. „It Ain't Over Till It's Over... but It's Never Over — Emerging and Reemerging Infectious Diseases“. In: *New England Journal of Medicine* 387.22 (Dez. 2022), S. 2009–2011. ISSN: 0028-4793. DOI: 10.1056/NEJMp2213814.

[25] Beth A. Fischer und Michael J. Zigmond. „The Essential Nature of Sharing in Science“. In: *Science and Engineering Ethics* 16.4 (1. Dez. 2010), S. 783–799. ISSN: 1471-5546. DOI: 10.1007/s11948-010-9239-x.

[26] Freimut Bodendorf. *Daten- und Wissensmanagement*. Springer-Lehrbuch. Berlin/Heidelberg: Springer-Verlag, 2006. ISBN: 978-3-540-28743-8. DOI: 10.1007/3-540-28682-9.

[27] *G20 Leaders' Communique Hangzhou Summit*. European Commission - European Commission. 5. Okt. 2016. URL: https://ec.europa.eu/commission/presscorner/detail/en/STATEMENT_16_2967 (besucht am 09. 08. 2023).

[28] Aaron Gidding et al. „ArchaeoSTOR: A Data Curation System for Research on the Archeological Frontier“. In: *Future Generation Computer Systems*. Including Special Sections: Advanced Cloud Monitoring Systems & The Fourth IEEE International Conference on e-Science 2011 — e-Science Applications and Tools & Cluster, Grid, and Cloud Computing 29.8 (1. Okt. 2013), S. 2117–2127. ISSN: 0167-739X. DOI: 10.1016/j.future.2013.04.007.

[29] Sean Gillies et al. *Rasterio: Geospatial Raster I/O for Python Programmers*. Mapbox, 2013–.

[30] J. Gray et al. „Data Cube: A Relational Aggregation Operator Generalizing GROUP-BY, CROSS-TAB, and SUB-TOTALS“. In: *Proceedings of the Twelfth International Conference on Data Engineering*. Proceedings of the Twelfth International Conference on Data Engineering. Feb. 1996, S. 152–159. DOI: 10.1109/ICDE.1996.492099.

[31] Trisha Gura. „Citizen Science: Amateur Experts“. In: *Nature* 496.7444 (7444 Apr. 2013), S. 259–261. ISSN: 1476-4687. DOI: 10.1038/nj7444-259a.

[32] Elise Haak et al. „A Framework for Strengthening Data Ecosystems to Serve Humanitarian Purposes“. In: *Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age*. Dg.o '18. New York, NY, USA: Association for Computing Machinery, 30. Mai 2018, S. 1–9. ISBN: 978-1-4503-6526-0. DOI: 10.1145/3209281.3209326.

[33] W. Harrison. „Eating Your Own Dog Food“. In: *IEEE Software* 23.3 (Mai 2006), S. 5–7. ISSN: 1937-4194. DOI: 10.1109/MS.2006.72.

[34] Mohamed Helmy, Alexander Crits-Christoph und Gary D. Bader. „Ten Simple Rules for Developing Public Biological Databases“. In: *PLOS Computational Biology* 12.11 (10. Nov. 2016), e1005128. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1005128.

[35] Maurice Herlihy und Nir Shavit. *The Art of Multiprocessor Programming*. Revised first edition. Amsterdam: Morgan Kaufmann, 2012. 508 S. ISBN: 978-0-12-397337-5.

[36] Peter J. Hotez. „Southern Europe's Coming Plagues: Vector-Borne Neglected Tropical Diseases“. In: *PLoS Neglected Tropical Diseases* 10.6 (30. Juni 2016), e0004243. ISSN: 1935-2727. DOI: 10.1371/journal.pntd.0004243.

[37] Peter J. Hotez und Aruna Kamath. „Neglected Tropical Diseases in Sub-Saharan Africa: Review of Their Prevalence, Distribution, and Disease Burden“. In: *PLoS Neglected Tropical Diseases* 3.8 (25. Aug. 2009), e412. ISSN: 1935-2727. DOI: 10.1371/journal.pntd.0000412.

[38] Dean T Jamison et al. *Disease Control Priorities in Developing Countries, Second Edition*. World Bank and Oxford University Press, 2006. ISBN: 978-0-8213-6179-5.

[39] Muhammad Mahdi Karim. *Aedes Aegypti Feeding in Dar Es Salaam, Tanzania*. 2009. URL: https://commons.wikimedia.org/wiki/File:Aedes_aegypti_feeding.jpg (besucht am 18. 08. 2023).

[40] Karl Fogel. *Producing Open Source Software: How to Run a Successful Free Software Project*. Second. O'Reilly Media, Jan. 2017.

[41] W. J. M. Knoben et al. „Community Workflows to Advance Reproducibility in Hydrologic Modeling: Separating Model-Agnostic and Model-Specific Configuration Steps in Applications of Large-Domain Hydrologic Models“. In: *Water Resources Research* 58.11 (2022), e2021WR031753. ISSN: 1944-7973. DOI: 10.1029/2021WR031753.

[42] Steve Kopp et al. „Achieving the Full Vision of Earth Observation Data Cubes“. In: *Data* 4.3 (3 Sep. 2019), S. 94. ISSN: 2306-5729. DOI: 10.3390/data4030094.

[43] Sandra MacFadyen et al. „Drowning in Data, Thirsty for Information and Starved for Understanding: A Biodiversity Information Hub for Cooperative Environmental Monitoring in South Africa“. In: *Biological Conservation* 274 (2022), S. 109736. ISSN: 0006-3207. DOI: 10.1016/j.biocon.2022.109736.

[44] Jane P. Messina et al. „The Current and Future Global Distribution and Population at Risk of Dengue“. In: *Nature Microbiology* 4.9 (9 Sep. 2019), S. 1508–1515. ISSN: 2058-5276. DOI: 10.1038/s41564-019-0476-8.

[45] Jennifer C. Molloy. „The Open Knowledge Foundation: Open Data Means Better Science“. In: *PLOS Biology* 9.12 (6. Dez. 2011), e1001195. ISSN: 1545-7885. DOI: 10.1371/journal.pbio.1001195.

[46] Carmen Morales et al. „Earth Map: A Novel Tool for Fast Performance of Advanced Land Monitoring and Climate Assessment“. In: *Journal of Remote Sensing* 3 (12. Jan. 2023), S. 0003. DOI: 10.34133/remotesensing.0003.

[47] Harry Mucksch und Wolfgang Behme, Hrsg. *Das Data Warehouse-Konzept*. Wiesbaden: Gabler Verlag, 2000. ISBN: 978-3-409-42216-1 978-3-322-89533-2. DOI: 10.1007/978-3-322-89533-2.

[48] Gaspary O. Mwanyika et al. „Dengue Virus Infection and Associated Risk Factors in Africa: A Systematic Review and Meta-Analysis“. In: *Viruses* 13.4 (4 Apr. 2021), S. 536. ISSN: 1999-4915. DOI: 10.3390/v13040536.

[49] John N. Nkengasong, Katy Yao und Philip Onyebujoh. „Laboratory Medicine in Low-Income and Middle-Income Countries: Progress and Challenges“. In: *The Lancet* 391.10133 (12. Mai 2018), S. 1873–1875. ISSN: 0140-6736, 1474-547X. DOI: 10.1016/S0140-6736(18)30308-8.

[50] Marcelo Iury S. Oliveira und Bernadette Farias Lóscio. „What Is a Data Ecosystem?“ In: *Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age*. Dg.o '18. New York, NY, USA:

Association for Computing Machinery, 30. Mai 2018, S. 1–9. ISBN: 978-1-4503-6526-0. DOI: 10.1145/3209281.3209335.

[51] Open Source Geospatial Foundation. *GeoNode's Documentation*. 2020. URL: https://docs.geonode.org/en/master/ (besucht am 23. 10. 2023).

[52] Albert D. M. E. Osterhaus et al. „Make Science Evolve into a One Health Approach to Improve Health and Security: A White Paper". In: *One Health Outlook* 2.1 (17. Apr. 2020), S. 6. ISSN: 2524-4655. DOI: 10.1186/s42522-019-0009-7.

[53] PAHO. *EWAR Concept and Implementation of EWARS-in-a-Box During Complex Emergencies - PAHO/WHO | Pan American Health Organization*. 7. Juli 2021. URL: https://www.paho.org/en/documents/ewar-concept-and-implementation-ewars-box-during-complex-emergencies (besucht am 08. 09. 2023).

[54] Melissa Lee Phillips. „Dengue Reborn: Widespread Resurgence of a Resilient Vector". In: *Environmental Health Perspectives* 116.9 (Sep. 2008), A382–A388. ISSN: 0091-6765.

[55] Jessica Pintado Silva und Ana Fernandez-Sesma. „Challenges on the Development of a Dengue Vaccine: A Comprehensive Review of the State of the Art". In: *Journal of General Virology* 104.3 (2023), S. 001831. ISSN: 1465-2099. DOI: 10.1099/jgv.0.001831.

[56] Elizabeth Pisani et al. „Time for Fair Trade in Research Data". In: *The Lancet* 375.9716 (27. Feb. 2010), S. 703–705. ISSN: 0140-6736, 1474-547X. DOI: 10.1016/S0140-6736(09)61486-0.

[57] Rufus Pollock. *Building the (Open) Data Ecosystem*. 31. März 2011. URL: https://blog.okfn.org/2011/03/31/building-the-open-data-ecosystem/ (besucht am 18. 10. 2023).

[58] Jonathan A. Polonsky et al. „Outbreak Analytics: A Developing Data Science for Informing the Response to Emerging Pathogens". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 374.1776 (20. Mai 2019), S. 20180276. DOI: 10.1098/rstb.2018.0276.

[59] Ignatius Ryan Pranantyo, Mahardika Fadmastuti und Fredy Chandra. „InaSAFE Applications in Disaster Preparedness". In: *AIP Conference Proceedings* 1658.1 (24. Apr. 2015), S. 060001. ISSN: 0094-243X. DOI: 10.1063/1.4915053.

[60] Erhard Rahm, Gunter Saake und Kai-Uwe Sattler. *Verteiltes und Paralleles Datenmanagement: Von verteilten Datenbanken zu Big Data und Cloud*. eXamen.press. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. ISBN: 978-3-642-45241-3 978-3-642-45242-0. DOI: 10.1007/978-3-642-45242-0.

[61] RfII – Rat für Informationsinfrastrukturen. *Datenpolitik, Open Science und Dateninfrastrukturen: Aktuelle Entwicklungen im europäischen Raum*. Göttingen, Juni 2022, S. 92.

[62] Richard Stallmann. „What Is the Free Software Foundation?“ In: *GNU Bulletins* 1.1 (Feb. 1986).

[63] Joacim Rocklöv und Robert Dubrow. „Climate Change: An Enduring Challenge for Vector-Borne Disease Prevention and Control“. In: *Nature Immunology* 21.5 (5 Mai 2020), S. 479–483. ISSN: 1529-2916. DOI: 10.1038/s41590-020-0648-y.

[64] Daniel Runfola et al. „geoBoundaries: A Global Database of Political Administrative Boundaries“. In: *PLOS ONE* 15.4 (24. Apr. 2020), e0231866. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0231866.

[65] Marcelo Iury S. Oliveira, Glória de Fátima Barros Lima und Bernadette Farias Lóscio. „Investigations into Data Ecosystems: A Systematic Mapping Study“. In: *Knowledge and Information Systems* 61.2 (1. Nov. 2019), S. 589–630. ISSN: 0219-3116. DOI: 10.1007/s10115-018-1323-6.

[66] Jeffrey S. Saltz und Neil Dewar. „Data Science Ethical Considerations: A Systematic Literature Review and Proposed Project Framework“. In: *Ethics and Information Technology* 21.3 (1. Sep. 2019), S. 197–208. ISSN: 1572-8439. DOI: 10.1007/s10676-019-09502-5.

[67] Geir Kjetil Sandve et al. „Ten Simple Rules for Reproducible Computational Research“. In: *PLOS Computational Biology* 9.10 (24. Okt. 2013), e1003285. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1003285.

[68] Matthias Scheffler et al. „FAIR Data Enabling New Horizons for Materials Research“. In: *Nature* 604.7907 (7907 Apr. 2022), S. 635–642. ISSN: 1476-4687. DOI: 10.1038/s41586-022-04501-x.

[69] Smisha Agarwal et al. *Digital Solutions for COVID-19 Response: An Assessment of Digital Tools for Rapid Scale-up for Case Management and Contact Tracing*. Johns Hopkins Global mHealth Initiative (JHU-GmI) - Johns Hopkins Bloomberg School of Public Health, 2020.

[70] Nattachai Srisawat et al. „World Dengue Day: A Call for Action“. In: *PLOS Neglected Tropical Diseases* 16.8 (4. Aug. 2022), e0010586. ISSN: 1935-2735. DOI: 10.1371/journal.pntd.0010586.

[71] Richard Stallman. „Why Öpen Source"Misses the Point of Free Software“. In: *Communications of the ACM* 52.6 (1. Juni 2009), S. 31–33. ISSN: 0001-0782. DOI: 10.1145/1516046.1516058.

[72] Daniel Joseph Sturtevant. „System Design and the Cost of Architectural Complexity“. Thesis. Massachusetts Institute of Technology, 2013.

[73] Sina Salajegheh Tazerji et al. „An Overview of Anthropogenic Actions as Drivers for Emerging and Re-Emerging Zoonotic Diseases“. In: *Pathogens* 11.11 (11 Nov. 2022), S. 1376. ISSN: 2076-0817. DOI: 10.3390/pathogens11111376.

[74] The pandas development team. *Pandas-Dev/Pandas: Pandas*.

[75] Michael A. Tolle. „Mosquito-Borne Diseases“. In: *Current Problems in Pediatric and Adolescent Health Care* 39.4 (1. Apr. 2009), S. 97–140. ISSN: 1538-5442. DOI: 10.1016/j.cppeds.2009.01.001.

[76] Ana Trisovic et al. „A Large-Scale Study on Research Code Quality and Execution“. In: *Scientific Data* 9.1 (1 21. Feb. 2022), S. 60. ISSN: 2052-4463. DOI: 10.1038/s41597-022-01143-6.

[77] Theo Vos et al. „Global Burden of 369 Diseases and Injuries in 204 Countries and Territories, 1990–2019: A Systematic Analysis for the Global Burden of Disease Study 2019“. In: *The Lancet* 396.10258 (17. Okt. 2020), S. 1204–1222. ISSN: 0140-6736, 1474-547X. DOI: 10.1016/S0140-6736(20)30925-9.

[78] Mark Walport und Paul Brest. „Sharing Research Data to Improve Public Health“. In: *The Lancet* 377.9765 (12. Feb. 2011), S. 537–539. ISSN: 0140-6736, 1474-547X. DOI: 10.1016/S0140-6736(10)62234-9.

[79] Mark D. Wilkinson et al. „The FAIR Guiding Principles for Scientific Data Management and Stewardship“. In: *Scientific Data* 3.1 (1 15. März 2016), S. 160018. ISSN: 2052-4463. DOI: 10.1038/sdata.2016.18.

[80] Jeannette M. Wing. „The Data Life Cycle“. In: *Harvard Data Science Review* 1.1 (3. Juli 2019). ISSN: 2644-2353, 2688-8513. DOI: 10.1162/99608f92.e26845b4.

[81] World Health Organization. *Early Warning Alert and Response in Emergencies: An Operational Guide*. World Health Organization, 2022, xiv, 189 p. 189 S. ISBN: 978-92-4-006358-7.

[82] World Health Organization. *EWARS in a Box: Quick Start Guide*. 2023. 29 S. ISBN: 978-92-4-006676-2.

[83] World Health Organization. *Rapid Risk Assessment of Acute Public Health Events*. 2012. 44 S. ISBN: 978-7-117-19850-9.

[84] Xiaorong Yang et al. „Global Burden for Dengue and the Evolving Pattern in the Past 30 Years“. In: *Journal of Travel Medicine* 28.8 (1. Dez. 2021), taab146. ISSN: 1708-8305. DOI: 10.1093/jtm/taab146.

A Anhang

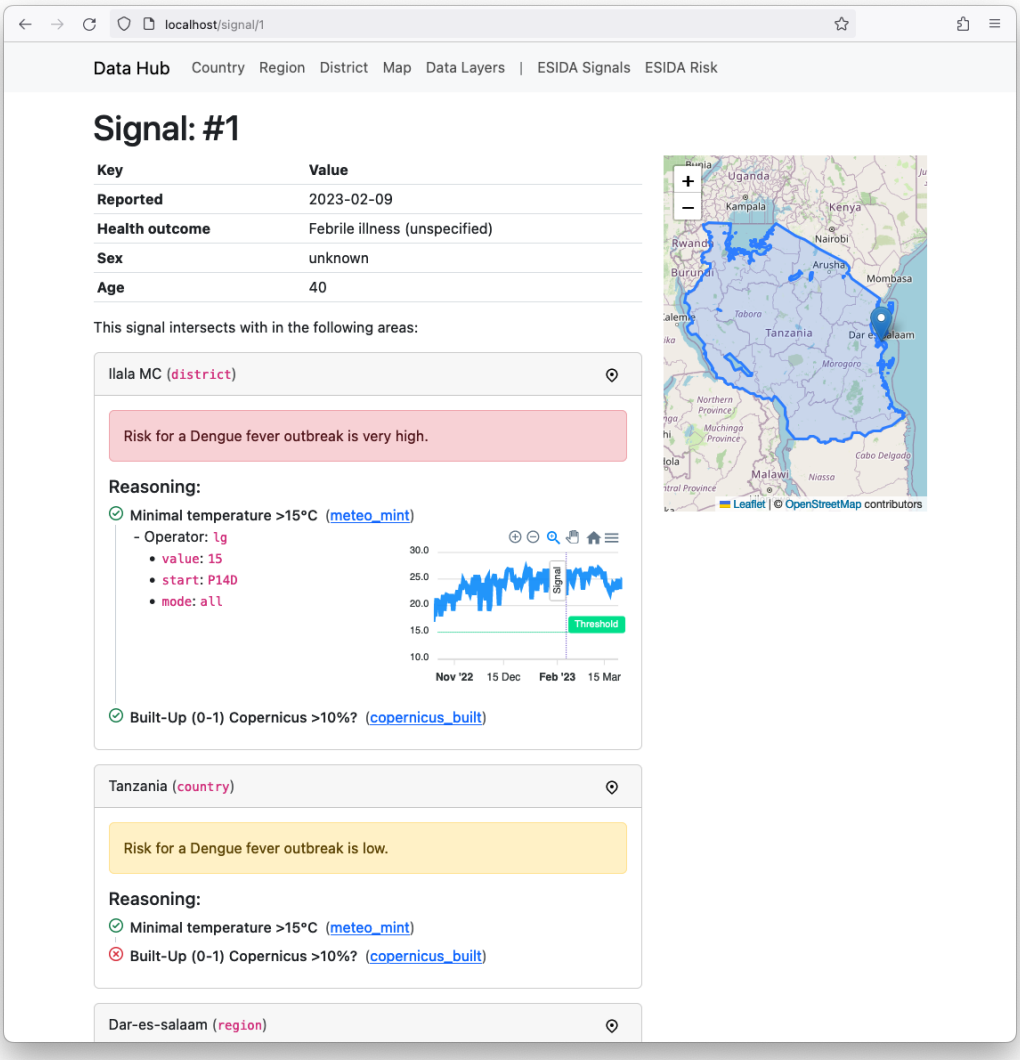


Abbildung A.1: Übersicht der Bewertung eines Signals.

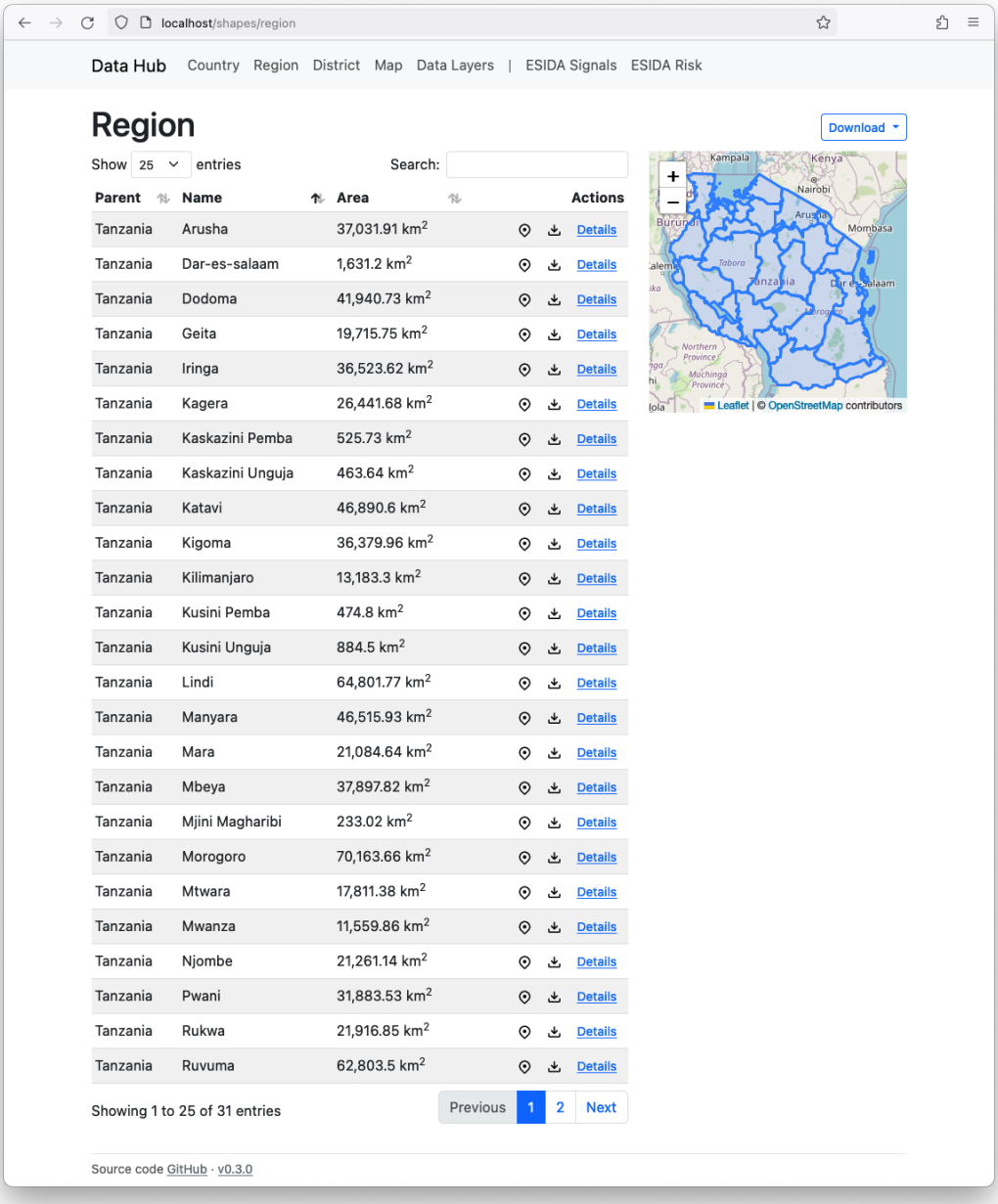


Abbildung A.2: Tabellarische Übersicht aller Shapes innerhalb einer Kategorie.

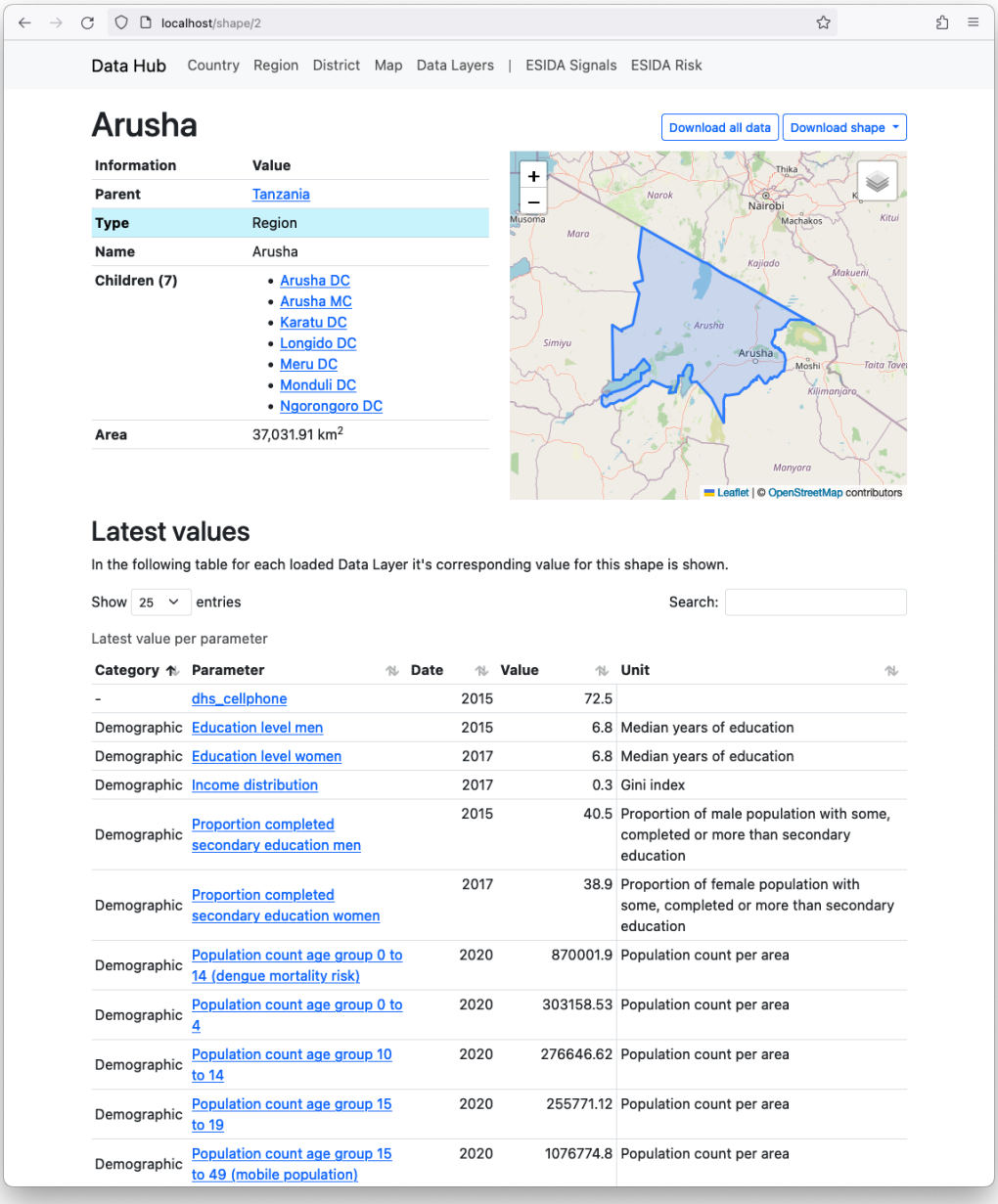


Abbildung A.3: Detailansicht eines einzelnen Shapes mit Data-Layer-Übersicht.

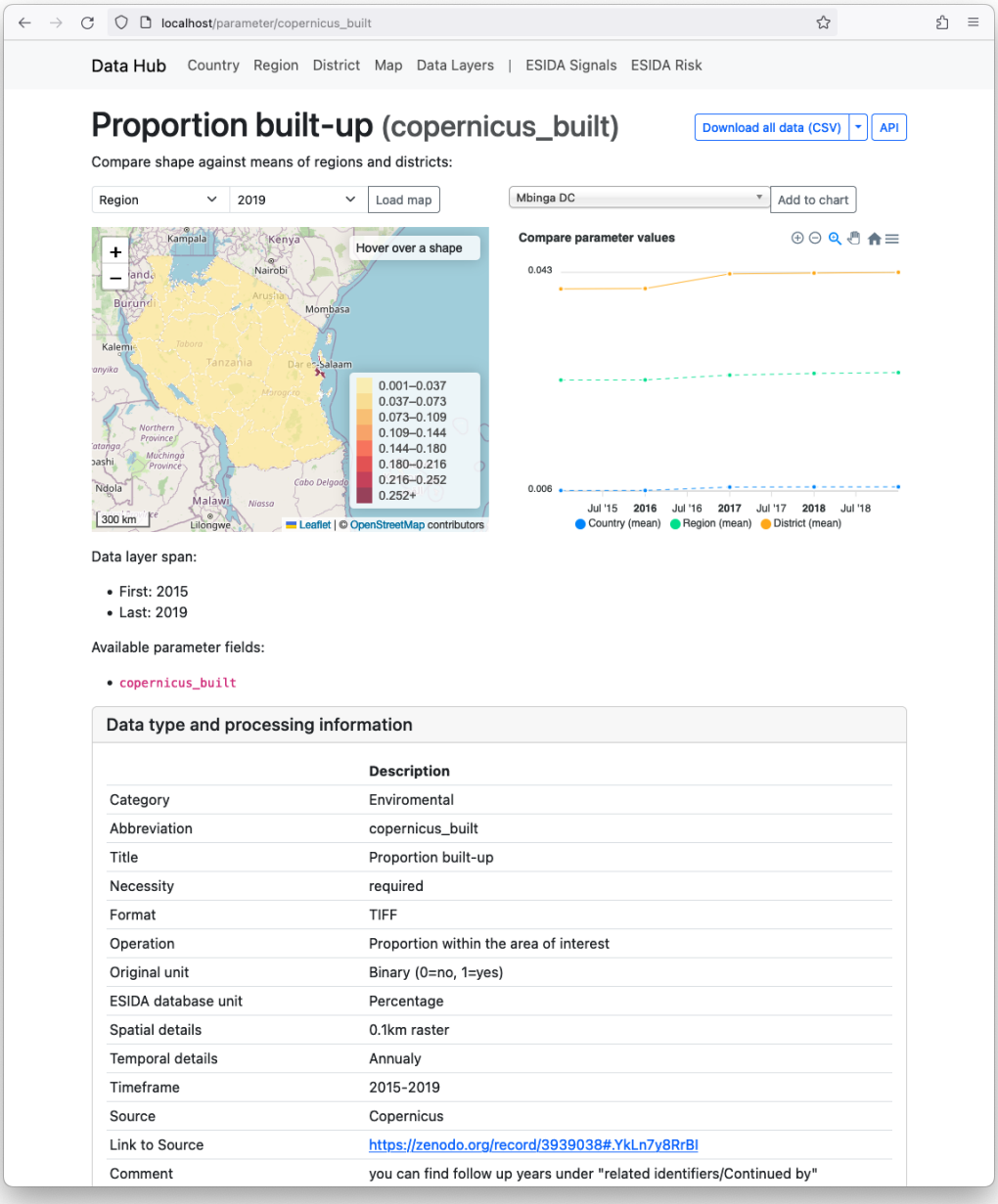


Abbildung A.4: Ansicht eines einzelnen Data Layers mit Analysemöglichkeiten.

Tabelle A.1: Auswahl der Metadatan der benötigten Data Layer.

Category	Abbreviation	Title	Format	Spatial details	Temporal details	Timeframe	License
Weather	chirps_tprecit	Total precipitation	TIFF	5km raster	Daily	1981-2020	CC BY 4.0
Weather	chirts_maxt	Maximum temperature	TIFF	5km raster	Daily	1983-2016	CC0 1.0
Weather	chirts_mint	Minimum temperature	TIFF	5km raster	Daily	1983-2016	CC0 1.0
Environmental	copernicus_built	Proportion built-up	TIFF	0.1km raster	Annualy	2015-2019	CC BY 4.0
Health	lit_conflabcap	Confirmatory diagnostic testing capacity	static	Country	Cross-sectional	2023	None
Health	lit_rdtcap	Rapid diagnostic testing capacity	static	Country	Cross-sectional	2023	None
Health	lit_veccontrol	Vector control capacity	static	Country	Cross-sectional	2023	None
Health	lit_vectabund	Vector abundance	static	Country	Cross-sectional	2023	None
Infrastructure	malariaatlas_traveltimehc	Average time to health care facility	TIFF	1km raster	Cross-sectional	2019	CC BY 3.0
Weather	meteo_maxt	Maximum temperature	API	Coordinates	Hourly	2010-present	CC BY-NC 4.0
Weather	meteo_mint	Minimum temperature	API	Coordinates	Hourly	2010-present	CC BY-NC 4.0
Weather	meteo_rhum	Relative humidity	API	Coordinates	Hourly	2010-present	CC BY-NC 4.0
Weather	meteo_tprecit	Total precipitation	API	Coordinates	Hourly	2010-present	CC BY-NC 4.0
Infrastructure	osm_airports	Airport infrastructure quality index	API	Coordinates	Cross-sectional	Most recent (2023)	ODbl AND Unknown
Infrastructure	osm_ferries	Ferry infrastructure quality index	API	Coordinates	Cross-sectional	Most recent (2023)	ODbl AND Unknown
Infrastructure	osm_rail	Rail infrastructure quality index	API	Coordinates	Cross-sectional	Most recent (2023)	ODbl AND Unknown
Infrastructure	osm_roads_regional	Road infrastructure quality index	API	Coordinates	Cross-sectional	Most recent (2023)	ODbl AND Unknown
Infrastructure	osm_roads_trunk	Trunkroad infrastructure quality index	API	Coordinates	Cross-sectional	Most recent (2023)	ODbl AND Unknown
Environmental	rcmrd_elev	Elevation	TIFF	0.03km raster	Cross-sectional	2018	ODbl
Demographic	dhs_gini	Income distribution	CSV	Region	Cross-sectional	2011 - 2017	DHS[a]
Health	dhs_mosnet	Insecticide treated bednet (ITN) use	CSV	Region	Cross-sectional	2017	DHS[a]
Demographic	dhs_propsecedu_men	Proportion completed secondary education men	CSV	Region	Cross-sectional	1991 - 2016	DHS[a]
Demographic	dhs_propsecedu_women	Proportion completed secondary education women	CSV	Region	Cross-sectional	1991 - 2016	DHS[a]
Infrastructure	dhs_sanitation	Proportion improved sanitation	CSV	Region	Cross-sectional	1991 - 2017	DHS[a]
Infrastructure	tnbs_garbage	Garbage collection	PDF	Region	Cross-sectional	2012	NBS[b]
Health	tnbs_medlabdens	Medical laborarity density	PDF	Region	Cross-sectional	2014	NBS[b]
Health	tnbs_nursmidwdens	Nurse and midwife density	PDF	Region	Cross-sectional	2014	NBS[b]
Health	tnbs_physdens	Physician density	PDF	Region	Cross-sectional	2014	NBS[b]
Health	who_hospbeddens	Hospital bed density	static	Country	Anually	2000-2018	CC BY-NC-SA 3.0 IGO
Demographic	worldpop_age_0_14	Population count age 0 to 14 (dengue mortality risk)	TIFF	0.1km raster	Anually	2000-2020	CC BY 4.0
Demographic	worldpop_age_15_49	Population count age 15 to 49 (mobile population)	TIFF	0.1km raster	Anually	2000-2020	CC BY 4.0
Demographic	worldpop_age_5_39	Population count age 5 to 39 (dengue morbidity risk)	TIFF	0.1km raster	Anually	2000-2020	CC BY 4.0
Demographic	worldpop_age_65	Population count age over 65 (dengue mortality risk)	TIFF	0.1km raster	Anually	2000-2020	CC BY 4.0
Demographic	worldpop_popc	Population count	TIFF	0.1km raster	Annualy	2000-2020	CC BY 4.0
Demographic	worldpop_popd	Population density	TIFF	1km raster	Annualy	2000-2020	CC BY 4.0
Demographic	worldpop_poverty_cons200	Proportion of population living <2.00$income	TIFF	1km raster	Cross-sectional	2010	CC BY 4.0

[a] https://dhsprogram.com/data/terms-of-use.cfm
[b] https://www.nbs.go.tz/nbs/takwimu/references/Licence-Agreement-NBS.pdf

Tabelle A.2: Quellen der benötigten Data Layer.

Abbreviation	Link to Source
chirps_tprecit	https://www.chc.ucsb.edu/data/chirps
chirts_maxt	https://www.chc.ucsb.edu/data/chirtsdaily
chirts_mint	https://www.chc.ucsb.edu/data/chirtsdaily
copernicus_built	https://zenodo.org/record/3939038#.YkLn7y8RrBI
lit_conflabcap	https://doi.org/10.1371/journal.pntd.0011289
lit_rdtcap	Expert knowledge
lit_veccontrol	Selected publications / DOIs: 10.5772/67109, 10.1186/s41182-021-00395-z
lit_vectabund	Selected publications / DOIs: 10.4314/tjs.v46i3.6, 10.1038/s41564-019-0440-7, 10.7554/eLife.08347, 10.1186/s41182-021-00395-z, 10.1080/20477724.2016.1182719, 10.5772/67109
malariaatlas_traveltimehc	https://malariaatlas.org/explorer/#/
meteo_maxt	https://dev.meteostat.net/python/stations.html
meteo_mint	https://dev.meteostat.net/python/stations.html
meteo_rhum	https://dev.meteostat.net/python/stations.html
meteo_tprecit	https://dev.meteostat.net/python/stations.html
osm_airports	https://www.theglobaleconomy.com/rankings/air_transport_infrastructure/
osm_ferries	https://www.theglobaleconomy.com/rankings/seaports_quality/
osm_rail	https://www.theglobaleconomy.com/rankings/railroad_quality/
osm_roads_regional	https://www.theglobaleconomy.com/rankings/roads_quality/
osm_roads_trunk	https://www.theglobaleconomy.com/rankings/roads_quality/
rcmrd_elev	https://opendata.rcmrd.org/datasets/tanzania-srtm-dem-30-meters/explore?location=-6.645890%2C36.133316%2C6.85
dhs_gini	https://www.statcompiler.com/en/
dhs_mosnet	https://www.statcompiler.com/en/
dhs_propsecedu_men	https://www.statcompiler.com/en/
dhs_propsecedu_women	https://www.statcompiler.com/en/
dhs_sanitation	https://www.statcompiler.com/en/
tnbs_garbage	https://www.nbs.go.tz/index.php/en/census-surveys/population-and-housing-census/164-2012-phc-tanzania-basic-demographic-and-socio-economic-profile%20S.%20202
tnbs_medlabdens	(a) https://africaopendata.org/dataset/dashibodi-ya-afya/resource/e73c7f34-ca45-4702-adf2-7a30b02d8e76, (b) https://africaopendata.org/dataset/dashibodi-ya-afya/resource/494e2c86-c98a-4a0a-a5e9-1d593bb592d0
tnbs_nursmidwdens	https://africaopendata.org/dataset/dashibodi-ya-afya/resource/494e2c86-c98a-4a0a-a5e9-1d593bb592d0
tnbs_physdens	https://africaopendata.org/dataset/dashibodi-ya-afya/resource/494e2c86-c98a-4a0a-a5e9-1d593bb592d0
who_hospbeddens	https://www.who.int/data/gho/data/indicators/indicator-details/GHO/hospital-beds-(per-10-000-population)
worldpop_age_0_14	https://hub.worldpop.org/geodata/listing?id=30
worldpop_age_15_49	https://hub.worldpop.org/geodata/listing?id=30
worldpop_age_5_39	https://hub.worldpop.org/geodata/listing?id=30
worldpop_age_65	https://hub.worldpop.org/geodata/listing?id=30
worldpop_popc	https://www.worldpop.org/geodata/listing?id=69
worldpop_popd	https://www.worldpop.org/project/categories?id=18
worldpop_poverty_cons200	https://www.worldpop.org/geodata/summary?id=1273

Erklärung zur selbstständigen Bearbeitung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort	Datum	Unterschrift im Original