

Web based Machine Learning for OpenEOcubes - WebMLOpenEO (Pflichtenheft)

Martin Kriger, Michael Brüggemann, Tim Ciroth, Leonard Wittrock

17. Januar 2024

Projekt:	Web based Machine Learning for OpenEOcubes - WebMLOpenEO
Auftraggeber:	Institut für Geoinformatik, Universität Münster
Auftragnehmer:	M3TL

Version	Datum	Autor(en)
1.0	06.11.2023	Martin Kriger, Maximilian Reiner, Michael Brüggemann, Tim Ciroth, Leonard Wittrock
1.1	10.12.2023	Martin Kriger, Maximilian Reiner, Michael Brüggemann, Tim Ciroth, Leonard Wittrock
1.2	17.01.2024	Martin Kriger, Michael Brüggemann, Tim Ciroth, Leonard Wittrock

Inhaltsverzeichnis

1	Zielbestimmung	1
2	Produkteinsatz	1
3	Produktbeschreibung	1
4	Funktionale Anforderungen	3
4.1	Data Ingest	3
4.2	Processes	4
4.3	Visualisation	5
4.4	Data Download	5
4.5	Deployment	6
5	Nichtfunktionale Anforderungen	6
5.1	Performance	6
6	Architektur	7
6.1	Systemarchitektur	7
6.1.1	Frontend-Server	7
6.1.2	Node-Backend	7
6.1.3	R-Backend	8
6.2	Demoprozess	8
6.3	Mockups	8
7	User Stories	8
8	Zugang	9
9	Zeitplan	9

1 Zielbestimmung

Das Hauptziel dieses Projektes besteht darin, einen openEO-Dienst zu entwickeln und zu implementieren, der die Klassifizierung von Fernerkundungsbildern auf Grundlage eines Trainingsdatensatzes mit dem Random-Forest Algorithmus ermöglicht. Hierbei sollen bestehende Lösungen wie openEO-Cubes verwendet und um neue Funktionen erweitert werden. Ein essenzieller Bestandteil des Projekts ist die Integration eines innovativen Prozesses in die vorhandenen openEO-Abläufe. Der von uns entwickelte Prozess erwartet einen Trainingsdatensatz, der Informationen zu bekannten Klassen und ihren Standorten enthält, die Spezifikation von Hyperparametern sowie einer Area of Interest (AOI), welche zu klassifizieren sein soll. Die Ergebnisse der Klassifizierung sollen daraufhin auf zwei Arten zugänglich gemacht werden. Als eine interaktive Visualisierung mittels einer web-basierten Browseranwendung und als Downloadmöglichkeit.

Das Projekt strebt an, eine benutzerfreundliche und leistungsstarke Plattform zu schaffen, die nicht nur die Klassifizierung von Fernerkundungsbildern ermöglicht, sondern auch eine intuitive Interaktion und den Zugang zu den Ergebnissen sowohl für die Forschungsgemeinschaft als auch für interessierte Nutzerinnen und Nutzer erleichtert.

2 Produkteinsatz

Die Zielnutzer sind Forscher oder Fachleute im Bereich Fernerkundung und räumliches Modellieren, die maschinelles Lernen für LULC-Klassifikationen nutzen möchten. Sie verlassen sich auf Sentinel-2-Satellitenbilder für ihre Projekte, beherrschen das Training und die Anwendung von Machine-Learning-Modellen und sind an groß angelegten Kartierungs- bzw. Modellierungsanwendungen interessiert, verfügen jedoch nicht über die Hardwarekomponenten, um maschinelles Lernen auf großen Datensätzen durchzuführen. Das Produkt soll auf den eigenen Servern zu betreiben sein, um technologische Barrieren für Benutzer zu minimieren, die ihre Modelle in alltäglichen Bildanalyseaufgaben ausführen und bewerten möchten. Dieser Dienst soll leicht anwendbar und flexibel sein und gewährleistet eine einfache Infrastruktur für ähnliche Anwendungsfälle.

3 Produktbeschreibung

Unser Produkt stellt einen umfassenden Webservice zur Verfügung, welcher aus einem Frontend-Server, einem Node.js-Backend und einem OpenEO-konformen R Backend besteht. Der Frontend Server stellt der Nutzer:in eine gerenderte HTML Seite zur Verfügung, auf der die Klassifikationsaufgabe konfiguriert werden kann. Der Frontendserver stellt ebenfalls eine HTML mit der Dokumentation zu unserem Webservice zur Verfügung. Durch das Verwenden von Vue.js ist außerdem eine umfangreiche Browserunterstützung gewährleistet (90%).

Im Node.js-Backend werden die Anfragen des Users gebündelt. Dort wird eine API bereitgestellt, die die Services unserer Anwendung beschreibt (API-self-documentation), sowie die Anfragen der Nutzer:innen verarbeitet, bzw. in neue Anfragen an andere Server umwandelt (Routing).

Die Anfragen zu einer Klassifikation werden dann an ein OpenEO-konformes Backend weitergeleitet, auf welchem die tatsächlichen Berechnungen stattfinden. Dieses nutzt die bereits bestehenden Strukturen von 'openEOcubes' und erweitert diese um geeignete Pro-

zesse, um die machine-learning gestützte Klassifikation durchzuführen. So werden bereits bestehende Prozesse zur Verarbeitung von Datacubes durch 'openEOcubes' genutzt und auch neue Prozesse im Bereich machine-learning ergänzt.

Der Webservice selbst wird auf DockerHub als Docker-Image bereitgestellt, welches direkt auf einer Amazon AWS/EC2 Instanz gehostet wird. So ist sichergestellt, dass der Service unabhängig vom lokalen System ausgeführt wird und der Webservice dadurch ebenfalls über die für die Klassifikationen nötige Rechenleistung verfügt. Dieses Docker Image besteht dabei aus 3 Services. Dies sind die oben beschriebenen Komponenten des Services (Frontend, Node.js-Backend, R Backend). Da jede Komponente nur eine einzelne Programmiersprache nutzt, kann auf komplexe Build-Prozesse mit Makefiles verzichtet werden. Unser Entwicklungsprozess besteht hier aus in den Testsuites 'Jest', 'vitest' und 'testthat' definierten Unit- und Integrations-Tests, sowie einem nach dem 4-Augenprinzip überwachten Pullrequest-Prozess in unserem Github-Repository.

Der gesamte Webservice wird unter einer OSI Lizenz veröffentlicht und steht als OpenSource-Software zur Verfügung. Der Kunde erhält über unser GitHub Repository während des Entwicklungsprozesses Zugriff auf den Quellcode, sowie alle dort verwalteten Projektmanagementdokumente (siehe Abschnitt 8).

4 Funktionale Anforderungen

4.1 Data Ingest

Nummer	Bezug	Beschreibung
/FA05/		Unsere Web-App wird als eine One-Page-Struktur implementiert, welche auf der Hauptseite ein Formular besitzt, um die benötigten Daten für die Verarbeitung einzugeben und einzuladen.
/FA10/	support loading training data via direct file upload to the server, support at least file formats GeoJSON and GeoPackage	Im Formular wird ein Button bereitgestellt, der den File-Explorer öffnet, um Dateien für die Trainingsdaten hochzuladen. GeoJSON und GeoPackage werden unterstützt.
/FA15/	the server automatically discovers and processes (accesses, or downloads) the satellite imagery from other sources as required for model training and prediction based on the locations and extents of the area of interest and training data	Mithilfe einer STAC API werden die Satellitenbilder, welche auf einem S3 Bucket bei AWS liegen angefragt, sodass entwickelte Prozessgraphen mit übergebenen Argumenten zur AOI und dem Trainingsgebiet weitere Argumente nutzen können. Hierfür verwenden wir die R Bibliothek 'rstac'.
/FA20/	the server automatically discovers and processes (accesses, or downloads) the satellite imagery from other sources as required for model training and prediction based on the locations and extents of the area of interest and training data, /F05/	Durch die gleichzeitige und gemeinsame Speicherung auf einer AWS-Instanz (us-west-2 region Oregon) wird der direkte Zugriff auf die Collection der Erdbeobachtungsdaten des Sentinel-2 Satelliten möglich.
/FA25/	example satellite data for demonstration purposes is being kept available on the server	Unsere Anwendung bietet die Möglichkeit über einen Button einen Demoprozess zu starten. Dies ist ein geführter Prozess, indem die Nutzer:in eine mögliche Eingabe in unseren Webservice vorgestellt bekommt und diese Anfrage dann an den Server gesandt wird.

4.2 Processes

Nummer	Bezug	Beschreibung
/FA30/	implement at least one standard machine learning algorithm for LULC classification supporting training and prediction on data cubes holding Sentinel-2 satellite image data	Wir implementieren den 'Random Forest' Algorithmus. Random Forest zeichnet sich durch eine Hohe Performanz in LULC Klassifikationsproblemen aus und ist daher gut für diese Anwendung geeignet
/FA35/	allow the user to specify the target area by means of polygon data (e.g.by file upload, or on-screen digitizing).	Der Nutzer wird die Möglichkeit haben in einer Karte, welche auf openLayers basiert, mithilfe eines Zeichenwerkzeuges, einen Kartenausschnitt mit der Area of Interest (AOI) einzuzeichnen.
/FA40/	allow the user to specify the target area by means of polygon data (e.g. by file upload, or on-screen digitizing)	Optional kann eine GeoJSON-Datei, welche den gewünschten Bereich als beliebiges Polygon definiert, über einen Button, welcher den File-Explorer öffnet, hochgeladen werden.
/FA45/	allow the user to specify the target area (rectangular) and date	Neben der Möglichkeit einen Kartenausschnitt zu bestimmen, ist es möglich im Formular über eine Datumsauswahl, den Zeitpunkt für die gewünschten Satellitendaten, festzulegen.
/FA55/	allow the user to specify a lower resolution to compute predictions on than the native resolution of the satellite imagery used	Wir stellen ein Input Feld zur Verfügung, in der die Resolution des Satellitenbildes eingestellt werden kann. Im Backend wird dies durch 'openEOcubes' in Verbindung mit 'gdalcubes' realisiert.
/FA60/	in addition to the classified map, create and return the map with class probabilities of all classes of that for which the probability is maximum.	Nach der Prozessierung wird dem Nutzer, über die Interaktive Karte, eine Möglichkeit gegeben die verschiedenen Kartentypen (predicted map, map with class probabilities) als Layerwechsel anzeigen zu lassen.
/FA65/	improve openEOcubes and submit one or more pull requests to its GitHub repo	Die Erweiterung wird ein neu definierter openEO-Prozess sein, der das Berechnen einer LULC-Klassifikation mit dem Algorithmus 'Random Forest' ermöglicht und dabei auf dem Backend laufen wird. Zusätzlich dazu werden die Teilprozesse einer ML-Klassifikation als openEO-konforme Prozesse implementiert.

4.3 Visualisation

Nummer	Bezug	Beschreibung
/FA70/	visualize the training areas/locations (if applicable) and area of interest on an interactive map (leaflet, openlayers or similar)	Die Web-App verfügt über eine Kartenansicht, die je nach ausgewähltem Schritt entweder die 'Area of Interest' oder die Training-Areas anzeigt.
/FA75/	visualize the resulting prediction (classification) on an interactive web map	Nach Abschluss des Prozesses zeigt die Karte das Ergebnis in Form eines ein- und ausblendbaren Klassifizierungs-Layers an. Dieser Layer verwendet eine barrierefreie Farblegende, die auf Grundlage der vom Backend gelieferten Daten berechnet wird. Die Farblegende ermöglicht den Benutzern eine klare und zugängliche Darstellung der Klassifizierungsinformationen auf der Karte, wodurch sie die Ergebnisse leicht verstehen können.
/FA80/	visualize the resulting quality of prediction (e.g. probability of the most likely class) on an interactive web map	Am Ende des Prozesses hat der Nutzer die Möglichkeit, eine Karte mit den Klassenwahrscheinlichkeiten in der Web-App anzuzeigen.

4.4 Data Download

Nummer	Bezug	Beschreibung
/FA85/	enable the download of the prediction map in a standard raster data format	Unsere Web-App bietet einen Download-Button an, der es den Benutzern ermöglicht, die Rasterdaten im .TIFF-Format herunterzuladen.
/FA90/	enable download of the trained model (if applicable)	Es wird eine Möglichkeit geben, das trainierte Model im '.rds' Format herunterzuladen.

4.5 Deployment

Nummer	Bezug	Beschreibung
/FA95/	Docker/docker-compose configurations are used for packaging and deployment	Das Projekt besteht aus drei Docker-Services: ein HTTP-Server mit Vue.js, ein Node.js-Server und ein R-Server mit openEOcubes. Diese Services sind für die Bereitstellung der Benutzeroberfläche und serverseitige Logik verantwortlich.
/FA100/	the developed system can be deployed to any host system without additional software requirements other than recent versions of Docker/docker-compose	Das Projekt kann mit Docker Compose gestartet werden, was die einfache Verwaltung und Bereitstellung der Anwendung ermöglicht.
/FA105/	eigene Implementierung	Alle Endpunkte unserer API im Node-Backend werden OpenAPI konform implementiert sein. Dies hat den Vorteil, dass unser Service nicht nur über die Website (als GUI) bedienbar ist, sondern auch direkt aus anderen Clients (Python, R, JS) per HTTP-Requests genutzt werden kann.
/FA110/	eigene Implementierung	Wir integrieren eine technische Umsetzung einer Swagger API-Dokumentation für unser Projekt. Obwohl dies nicht explizit in den anfänglichen Anforderungen aufgeführt war, soll diese Dokumentation die technische Umsetzung unserer Endpunkte zeigen und detaillierte Einblicke in die zugrundeliegende Implementierung bieten.

5 Nichtfunktionale Anforderungen

Die meisten nichtfunktionalen Anforderungen werden bereits in dem Abschnitt 'Produkt-Umgebung' dargestellt (siehe Abschnitt 3). Folglich wird hier weiterführend auf die Performance eingegangen.

5.1 Performance

Eine Änderung auf der Website erfolgt innerhalb einer Sekunde. Alle Inhalte werden asynchron geladen. Die interaktive Bedienung erfolgt innerhalb 0.1 Sekunden.

6 Architektur

6.1 Systemarchitektur

Unsere Applikation nutzt eine Kombination aus einem Frontend-Server, sowie zwei Backend-Servern (Node-Backend und R-Backend) (vgl. Abb. 1, im Anhang).

6.1.1 Frontend-Server

Der Frontend-Server stellt die HTML Seiten für die Nutzer:innen zur Verfügung, sowie einem Formular um mit der API unserer Applikation zu interagieren. Dort kann die Nutzer:in folgende Angaben machen:

- | | |
|----------------------------------|-----------------------------------|
| 1. 'machine-learning' (ML) Model | 4. Trainingsdaten |
| 2. 'Area of Interest' | 5. Hyperparameter |
| 3. 'Time of Interest' (TOI) | 6. Auflösung der Satellitenbilder |

Die Angabe des Models wählt aus, welches ML Model zur Klassifikation genutzt werden soll.

Die AOI sowie die Trainingsdaten kann die Nutzer:in über eine interaktive Karte angeben. Trainingsdaten können alternativ als GeoJSON oder Geopackage direkt hochgeladen werden. Die dort erzeugten Geometrien werden dann zu einer GeoJSON codiert und mit dem Formular übergeben.

Die TOI bestimmt ein Zeitintervall, in welchem die Sentinel-2 Satellitendaten herangezogen werden sollen.

Werden Hyperparameter für das Modell angegeben, so werden diese direkt für das zu trainierende ML Model verwendet. Da diese Parameter ggf. nicht optimal sein könnten, wird der Nutzer:in eine entsprechende Warnung gegeben. Werden keine Hyperparameter übergeben, so werden im R-Backend entsprechende Tuning-Methoden angewandt, um die optimalen Parameter zu ermitteln.

Mit der Auflösung kann bestimmt werden, wie groß ein Pixel des Satellitenbildes sein soll. Je feiner die gewählte Auflösung, desto präzisere Klassifikation sind möglich, aber auch der Klassifikationsaufwand ist deutlich erhöht.

6.1.2 Node-Backend

Das Node-Backend stellt die API zur Verfügung und verwaltet Anfragen an diese (über den Browser einer Nutzer:in oder einen HTTP-Request aus einem Client (R, JS, Python, etc.)). Dort werden die Requests validiert und weiterverarbeitet. Die Weiterverarbeitung erfolgt darin, dass die im HTTP-Request enthaltenen Parameter genutzt werden, um über den JS-Client von 'OpenEO' eine Anfrage an das R-Backend zu stellen. Auf dem Node-Backend werden sog. Prozessgraphen definiert, die das R-Backend anweisen, eine bestimmte Abfolge von Berechnungen durchzuführen. Die Ergebnisse des Prozesses werden dann vom Node-Backend an den Frontend-Server weitergereicht, der diese wiederum an den Client (Browser) weitergibt.

6.1.3 R-Backend

Das R-Backend basiert auf der Arbeit von Brian Pondi mit dem 'openEOcubes' Projekt. Wir verwenden seine Arbeit und entwickeln diese für unsere Applikation weiter. In 'openEOcubes' werden die Prozessgraphen entgegen genommen und decodiert. Hier werden eine Reihe von Prozessen in Gang gesetzt, um das im Prozessgraphen definierte Vorgehen durchzuführen. Am Ende des Prozesses werden die Ergebnisse an das Node-Backend zurückgegeben (in welchem Format die Daten übertragen werden, ist aus technischen Gründen bisher unklar). Die im Prozessgraphen übergebenen Argumente zur AOI und dem Trainingsgebiet (TrG) werden mit den weiteren Argumenten genutzt, um mittels einer STAC API Satellitenbilder anzufragen. Hierfür verwenden wir die R Bibliothek 'rstac'. Diese Satellitenbilder werden dann über die R-Bibliothek 'gdalcubes' zu Databricks zusammengefasst. Diese werden genutzt, um mittels der Geometrie der AOI und dem 'Trainingsgebiet' (TrG) Features für die ML Klassifikation aus den Databricks zu extrahieren (es wird ein Databrick für das Trainingsgebiet und ein Databrick für die AOI erstellt). Die so gewonnenen Features in Form eines R 'data.frame' können dann genutzt werden, um ein ML-Modell zu trainieren. Dieses wird dann einerseits direkt auf die Daten, die mittels der AOI extrahiert wurden, angewendet, als auch der Nutzer:in zum Download bereitgestellt.

6.2 Demoprozess

In einem Demoprozess (vgl. Abb. 2, im Anhang) wird Nutzer:innen der durch den Frontend-Server bereitgestellten HTML Seiten ermöglicht, unseren Service kennenzulernen. In einem geführten Prozess werden die einzelnen Felder des Formulars erklärt und weitere Informationen, warum diese Informationen nötig sind werden geliefert. Der Demoprozess endet dann damit, dass entweder eine Klassifikation, basierend auf den in der Formularmaske eingetragenen Daten erfolgen kann oder der Prozess durch die Nutzer:in abgebrochen wird.

Wird die Klassifikation angestoßen, so läuft der Prozess wie in Abschnitt 6.1 ab. Wird der Prozess abgebrochen, so wird die Nutzer:in wieder auf die Hauptseite unseres Service geleitet (leeres Formular).

Um die Prozessierungszeit zu minimieren, wird die Anfrage bereits vollkommen durch uns vorgegeben. So können wir sicherstellen, dass der Prozess maximal die geforderten 20 Sekunden andauert, während kein neuer Prozess für eine Demo auf dem R-Backend definiert werden muss.

6.3 Mockups

Mockups von unserer Hauptseite befinden sich im Anhang.

7 User Stories

- Als Nutzer:in möchte ich anhand einer interaktiven Benutzeroberfläche auf einer Karte den Kartenausschnitt wählen können, sodass ich den für mich interessanten Bereich untersuchen kann.
- Als Nutzer:in muss ich in der Lage sein das Datum zu bestimmen, Trainingsdaten über ein File hochladen zu können und einen Machine Learning Algorithmus mit

Hyperparametern auswählen zu können, damit ich mein eigenes individuelles Modell trainieren kann.

- Als Stadt Münster möchten wir, dass das Hochladen von Trainingsdaten auf jeden Fall die Formate GeoJSON und GeoPackage umfasst, damit wir die meisten unserer Daten direkt nutzen können.
- Als Nutzer:in möchte ich eine Möglichkeit haben neue Trainingsdaten zu bestimmen, um den Algorithmus wichtige Informationen zu geben, sodass eine möglichst umfangreiche Vorhersage durch den 'Random Forest' Algorithmus stattfinden kann.
- Als Laie und unaffine Person im Umgang mit Erdbeobachtungsdaten möchte ich eine Hilfestellung besitzen, die mir zeigt, wie ich vorgehen muss, um ein Modell trainieren zu können.
- Als wissenschaftliche:r Mitarbeiter:in einer Universität möchte ich eine Möglichkeit haben, die resultierenden Vorhersage mittels einer interaktiven Webkarte zu begutachten und als Rasterdaten zu downloaden und das trainierte Modell herunterzuladen, sodass ich diese weiterführend in meine Arbeit einbinden kann.
- Als Arbeitsgruppe an der Universität Osnabrück im Bereich Remote Sensing wollen wir eine Möglichkeit besitzen, das Toolkit mittels Docker auf verschiedensten Endgeräten laufen zu lassen, damit auch im Außendienst, spontane Analysen getätigt werden können.
- Als farbenblinder Mensch möchte ich barrierefrei und ohne Einschränkungen die Anwendung nutzen können, damit bei der Zusammenarbeit mit meinen Mitmenschen, die gleichen Voraussetzungen für alle bestehen und keine Fehlinterpretationen untereinander entstehen.

8 Zugang

Der Quellcode sowie die Projektmanagement-Dokumente sind auf unserem GitHub-Repository unter folgendem Link zu finden:

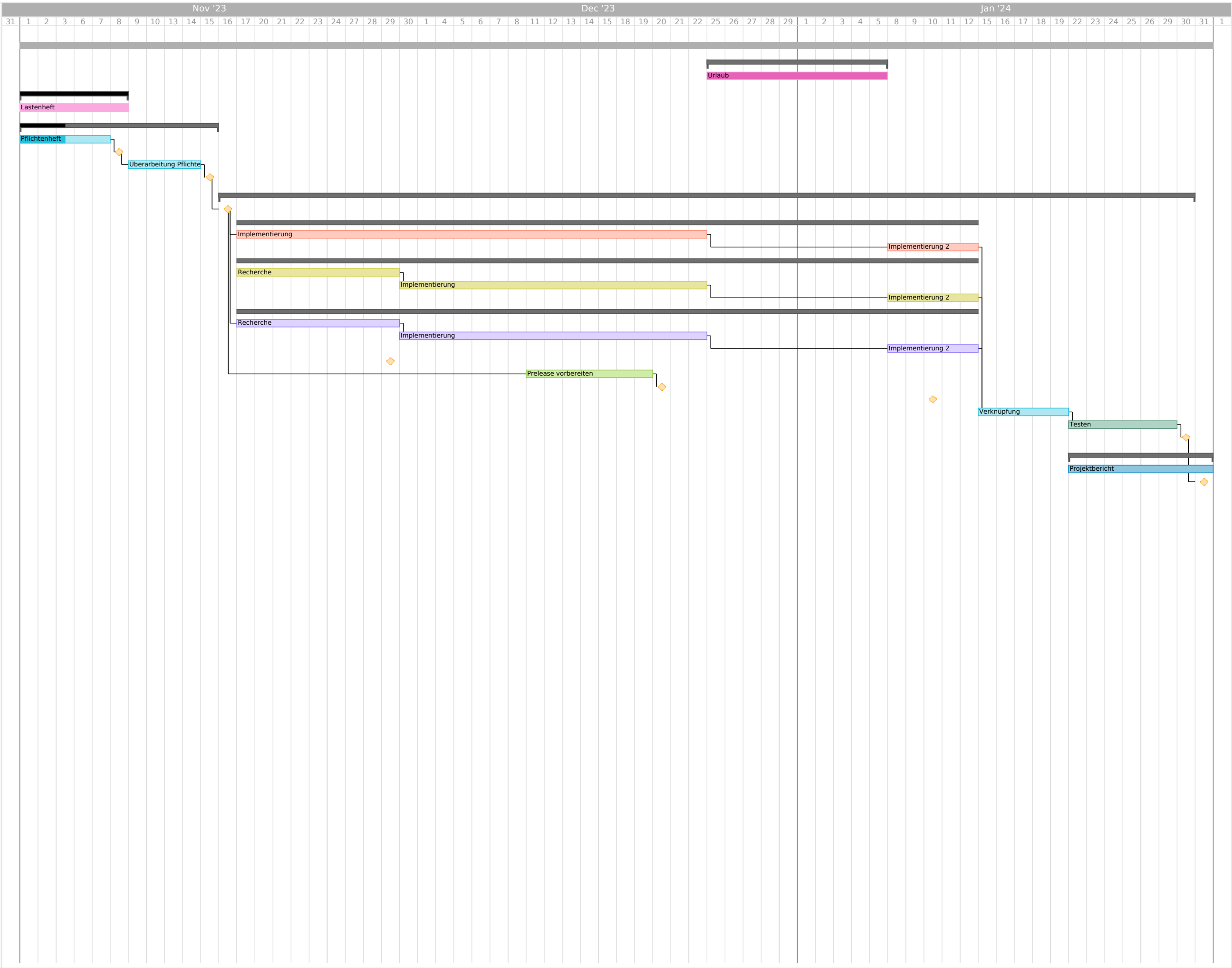
- [GitHub](#)

9 Zeitplan

Unsere voraussichtliche Zeiteinteilung befindet sich als Gantt-Chart im Anhang.

Geosoft2

- Ferien
- Urlaub
- Angebotsphase
- Lastenheft
- Planung
- Pflichtenheft
- Abgabe Pflichtenheft
- Überarbeitung Pflichtenheft
- Abgabe Pflichtenheft
- Implementierung
- Beginn Implementierung
- Frontend
- Implementierung
- Implementierung 2
- ML
- Recherche
- Implementierung
- Implementierung 2
- Node
- Recherche
- Implementierung
- Implementierung 2
- Status Report
- Prelease vorbereiten
- Pre-release
- Status Report
- Verknüpfung
- Testen
- Abgabe Endprodukt
- Projektabschluss
- Projektbericht
- Abschlusspräsentation



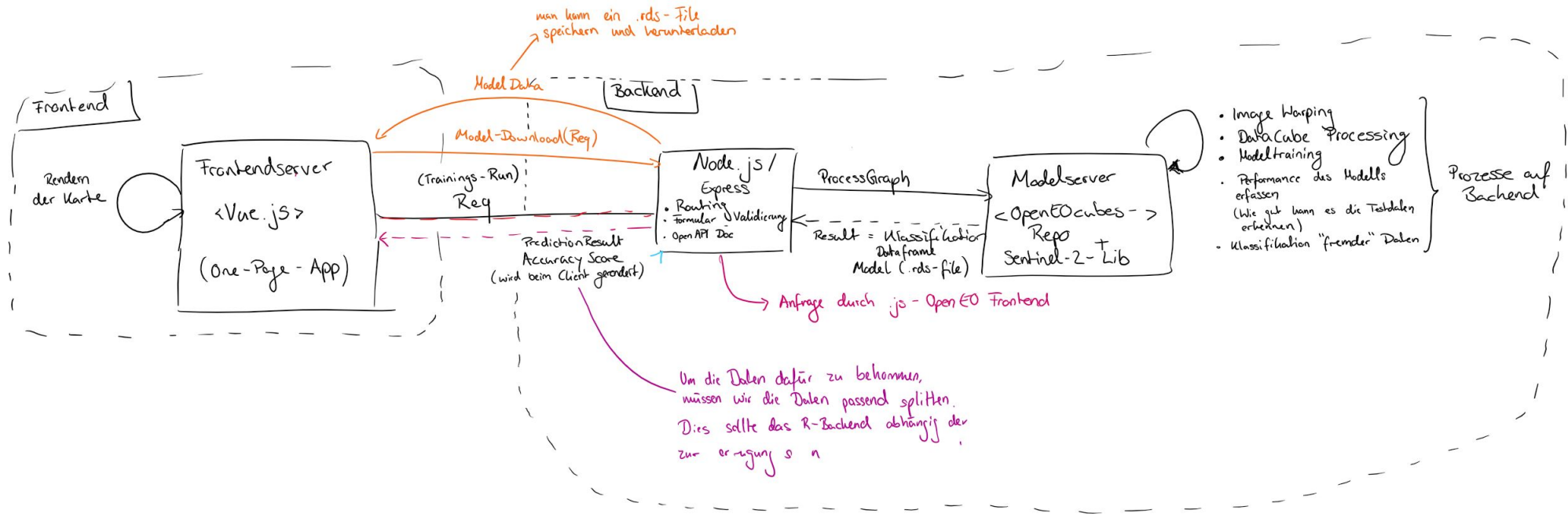


Abbildung 1: Systemarchitektur der Applikation

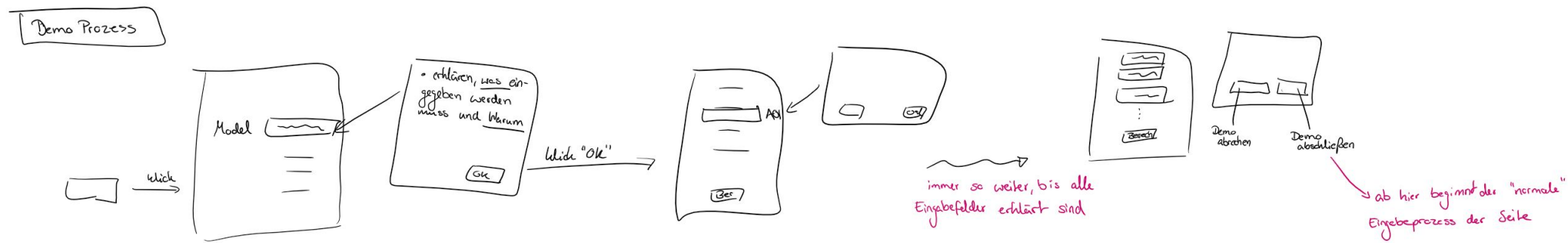


Abbildung 2: Demoprozess einer Klassifikation

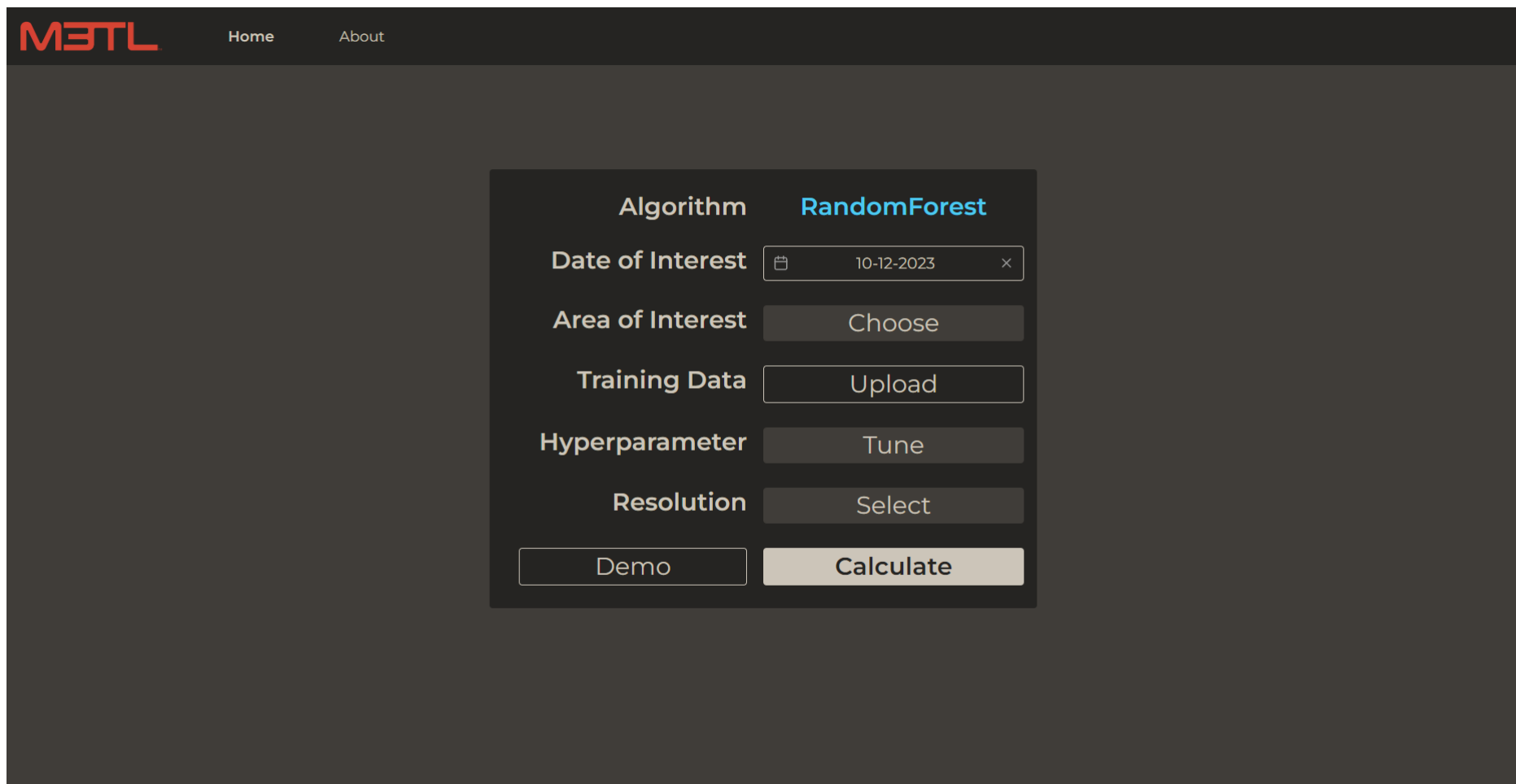


Abbildung 3: Die Hauptseite

Algorithm

RandomForest

Date of Interest

10-12-2023

×

Area of Interest

Training Data

Hyperparameter

Resolution

Demo

Calculate

<

Dec

2023

>

Mo

Tu

We

Th

Fr

Sa

Su

27

28

29

30

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

10-12-2023

Cancel


Select

Abbildung 4: Datumsauswahl

Area of Interest

+

−



Select on map

or

Upload GeoJSON

Cancel

Submit

Abbildung 5: Area of Interest