TOWARDS A NATIONAL DATA CUBE OF SATELLITE EARTH OBSERVATION DATA FOR ECOLOGICAL MODELING AND MONITORING

Nuno Filipe Escaleira de Sousa

Master's degree in Bioinformatics and Computational Biology

Biology Department

2021/2022

Supervisors

Joaquim Mamede Alonso, Professor, IPVC

Co-supervisors

Rita Paula Almeida Ribeiro, Professor, FCUP

João Pradinho Honrado, Professor , FCUP

U.PORTO

FC FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Sworn Statement

I, Nuno Filipe Escaleira de Sousa, enrolled in the Master Degree Master in Bioinformatics and Computational Biology at the Faculty of Sciences of the University of Porto hereby declare, in accordance with the provisions of paragraph a) of Article 14 of the Code of Ethical Conduct of the University of Porto, that the content of this dissertation reflects perspectives, research work and my own interpretations at the time of its submission.

By submitting this dissertation, I also declare that it contains the results of my own research work and contributions that have not been previously submitted to this or any other institution.

I further declare that all references to other authors fully comply with the rules of attribution and are referenced in the text by citation and identified in the bibliographic references section. This dissertation does not include any content whose reproduction is protected by copyright laws.

I am aware that the practice of plagiarism and self-plagiarism constitute a form of academic offense.

Nuno Filipe Escaleira de Sousa

September 30th 2022

Acknowledgements

Em primeiro lugar, agradecer à minha família. À minha mãe por me manter a cabeça no sítio independentemente da situação ao longo de todo este percurso. Ao meu pai e ao Eduardo por terem estado a meu lado durante o projeto.

Em segundo lugar, à Marlene por ter estado diariamente disposta a ouvir problemas de tese e percursos universitários, independentemente do que fazia ou não sentido, obrigado por tudo.

Aos meus avós por tudo que fizeram e por terem permitido que eu me dedicasse completamente ao trabalho, com o apoio presencial e remoto ao longo do processo.

Ao Ricardo, Bernardo e José por terem sempre prontas as piadas e alívios mentais quando as coisas ficavam mais difíceis. A todo o resto que tornou quer o percurso universitário, quer a dissertação uma tarefa mais fácil, o meu sincero obrigado também.

Por fim, aos meus orientadores, Joaquim, João, Rita pela ajuda sempre disponível. Apesar de não ter sido tão pessoal como seria de esperar, tiveram sempre disponíveis para mim e para o projeto quando foi necessário. Além destas 3 pessoas, também queria agradecer ao Bruno Marcos e João Gonçalves por toda a ajuda com o projeto e com a aclimatação à área de estudo, que tornou todo o processo mais fácil. Sem os 5 seria impossível ter completado este projeto.

Todo o projeto foi desenhado, planeado e executado no contexto temático dos projetos de investigação Projecto SeverusPT (FCT) e (https://severus.pt/) e projeto PorBiota (https://www.porbiota.pt/).

FCT
Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

Resumo

O acompanhamento e adaptação aos desafios socioeconómicos e sociais globais exigem novas abordagens para a captura de dados espaciais, desenvolvimento de infraestruturas de dados espaciais, partilha de dados e descoberta de conhecimento. Neste sentido, as plataformas de Observação da Terra (EO) permitem recolher e processar grandes dados sobre as principais características estruturais e funcionais dos sistemas socio-ecológicos. O aumento exponencial da quantidade, diversidade e qualidade das bases de dados de referência e temáticas e, em especial, dos produtos primários e secundários de OE, levou ao desenvolvimento de cubos de dados. Um cubo de dados de observação terrestre é uma matriz multidimensional de séries temporais padronizadas de dados de imagem. Esta dissertação apresenta uma visão geral dos desafios e construção de cubos de dados, e possíveis áreas de aplicações, com especial atenção para programas de monitorização/modelação ecológica. Utilizando exemplos já funcionais como o Cubo de Dados Suíço como prova de conceito, propomos então uma proposta inicial de plano de desenvolvimento do Cubo de Dados Português, com base na nossa própria prova de conceito. Como resultado, obtivemos um protótipo de estrutura de cubos de dados, com dados de 10 anos para a área continental de Portugal, para 9 produtos primários e 4 produtos secundários. Também produzimos um caso de estudo para demonstrar as capacidades do fluxo de trabalho e construímos um protótipo de interface visual para uso público. Os testes de desempenho foram realizados resultaram em tempos de computação aceitável, permitindo testes de escalabilidade e stress do cubo de dados.

Palavras-chave: [Earth Observation, Satellite Imaging, Data Structures, Data Cubes, Ecological Monitoring]

Abstract

Tracking and adapting to the global socio-ecological and societal challenges call for novel approaches to spatial data capture, spatial data infrastructure development, (meta)data sharing and knowledge discovery. In this regard, Earth Observation (EO) platforms allow to collect and process big data on key structural and functional features of socio-ecological systems. The exponentially increasing of quantity, diversity and quality of reference and thematic databases and especially of primary and secondary-EO products have led to the development of data cubes. An Earth Observation data cube is a multi-dimensional array of standardized EO time series of image data. This dissertation presents an overview of the data cube challenges and construction, and possible applications areas, with a special regard for ecological monitoring/modelling programs. It then takes a dive into the use of OLAP such as data cubes as ways of dealing with the influx of “Big Data” from EO. Using already functional examples such as the Swiss Data Cube as proof of concept, we then propose an initial proposal of development plan of Portuguese Data Cube, building on our own proof-of-concept. As a result, we obtained a prototype data cube structure, with 10-year data for the Portugal continental area, for 9 primary and 4 secondary products derived from the primary ones. We also produced a case study to demonstrate the capabilities of the workflow and built a prototype visual interface for public use. The performance tests ran returned a desirable computing time, allowing for the scalability and stress testing of the data cube.

Keywords: [Earth Observation, Satellite Imaging, Data Structures, Data Cubes, Ecological Monitoring]

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1. The challenge of ecological monitoring for global sustainability

Present human sustainability and earth habitability in Anthropocene era implies developing spatially explicit digital data and knowledge management aiming systems and managing the complex, dynamic and adaptive socio-ecological systems[1]. The critical relevance of human activities in earth systems functioning poses special attention on the socio-economic impacts in natural conditions and resources as well as, in ecosystems functions and services[2].

These challenges promoted a (trans)disciplinary research and an increasing data intensive and open science development, namely associated to data management needs and advances of (spatial) information systems and global/national data infrastructures. Ecological and societal challenges call for infrastructure development, (meta)data sharing and knowledge discovery. In this regard, Earth Observation (EO) platforms allow the collection and processing of very large amounts of data on key ecological systems. The increasing quantity, diversity and quality of reference and thematic databases and especially of EO products have led to the development of “data cubes”.

Although focused on EO data dimensional arrays of standardized EO time series of image data, the concept’s flexibility of the approach allows other gridded data collections to be included and analyzed. Several initiatives have been developed for the analysis of rich EO data, such as the Open Data Cube (ODC), an open-source project specialized in the managing and analysis of large quantities of spatial data. In this project we will take different approach, using R as a base language and delineating a pipeline for the deployment of a data cube.

1.2. Earth Observation

The development of a new era of data recovery and treatment, referred to as “Digital Earth”, a term coined in 1998 by Al Gore, the US vice-president, is the basis for the worldwide evolution in terms of technological and electronic developments in the human and ecological/natural sciences[3]. Amongst them are frameworks such as the Geographic Information System and geospatial data infrastructures, responsible for the recovery, processing, and deployment of products based on new and historical data, to magnify the extent of the world’s knowledge and a better prediction capacity of its sustainability (Figure 1).

Amongst the previously established advances, one of the main projects established was the Earth Observation, defined as the gathering of information and data about the Earth while using remote sensing techniques. These techniques can be separated according to the equipment or sensor types, such as optical, radar detection, LiDAR (Laser Imaging Detection and Ranging), multi or uni-spectral, and split according to the vehicle for sensing, such as satellites, UAV’s or aircraft-based sensing.

The data images gathered from the remote sensing have four major components:

1) spatial, the size of each recorded pixel;
2) spectral, the wavelength of each recorded band;
3) radiometric, the amount of different radiation intensities that the sensor can distinguish.
4) temporal or time-scale resolution between sequential/regular data captures, responsible for the major development of time-series development and analysis[4];

Earth Observation and remote sensing techniques brought forth a computing bottleneck for the previously available infrastructures, assuming processing challenges considering data observations time-series. The diversity of remote sensors and vehicles used also brought a variety of measurements that was not dealt with before. This led Earth Observation into the age of “Big, Linked and Open Data”.

The “Big Data” posed a need for improvement of processing/analyzing algorithms and storage availability. While the storage problem has been tackled with the evolution of workstations and cloud computing, there is still permanent difficulty in analyzing data sets of huge size while still being able to make a logical claim of the results derived from them[5].

In terms of Volume, the data obtained from satellites is usually in the terabytes range, which is not feasible for regular desktops or even workstations. In terms of Variety, with the improvement of the instruments/satellites, the improved resolutions and measurements will lead to more specific observations, which require crossing with previous archives to generate an on-going record[6], and the ability to process data that may not be aligned in format, semantics or structures, requiring the use of multiple programming languages or database queries(SQL as an example).

Velocity, as with Volume, translates to the need of faster processing algorithms and software/hardware, due to the evolution of the data from images taken years apart to now being able to access images taken with hours of difference. In response, the research bodies and the EO research groups have used data warehouses and Online Analytical Processing (OLAP), such as data cubes, to tackle these problems.

This led to the creation of international guidelines and directories, most of the times government led, which indicate the need for a transition into making all the “Big Data” obtained from Earth Observation freely and openly accessible. Amongst these, mention goes to the Inspire Directive, the Open GeoSpatial Consortium and Data Cube Programs such as the Open Data Cube initiative.

While the definition of “data cubes” is still variable across the literature, as described by the Open Data Cube, Data Cubes are defined as time series multidimensional stack of spatially aligned analysis ready data. Geospatially, this allows for a time-series analysis of a predetermined space vector in the data cube (Figure 3 and 4). Despite their name, these data warehouses are a n-dimensional structure, not exclusive to n=3, where regular EO data series presents a critical and central role.

Data Cubes, as spatial data infrastructures, are mostly comprised of five different stages/components:

1) the metadata and data services component, related to the data modeling and interoperability;

2) the technological (Software and Hardware, Back and Front end) component, defining the structure which supports the Data Cube;

3) the user component, both internal users/developers and the external or “end-product” users, and their relations with procedures and responsibility;

4) standards component, related to data, technology, and user´s norms;

5) the policy component, referring to the utilization made from the data cube, regarding strategic option and governance decision;

In recent years, there has been a tendency into making the general idea of data cube transition into a spatial data cube. This comes from the notion that every set of a data can be defined by having a spatial feature or dimension [7]

In the EO context, data cubes tackle the processing and analysis problems brought by the new data, in the 20TB per day magnitude [8] . The major development brought by the data cubes is a transition from a spatial dimension storage model, as the traditional methods did, to the ability to order the imported data as a non-spatial axis, most commonly time-series, in addition to the set spatial axis. This makes it so the storage method is oriented to an easy access tailored to the user’s needs, including data management and analysis [9].

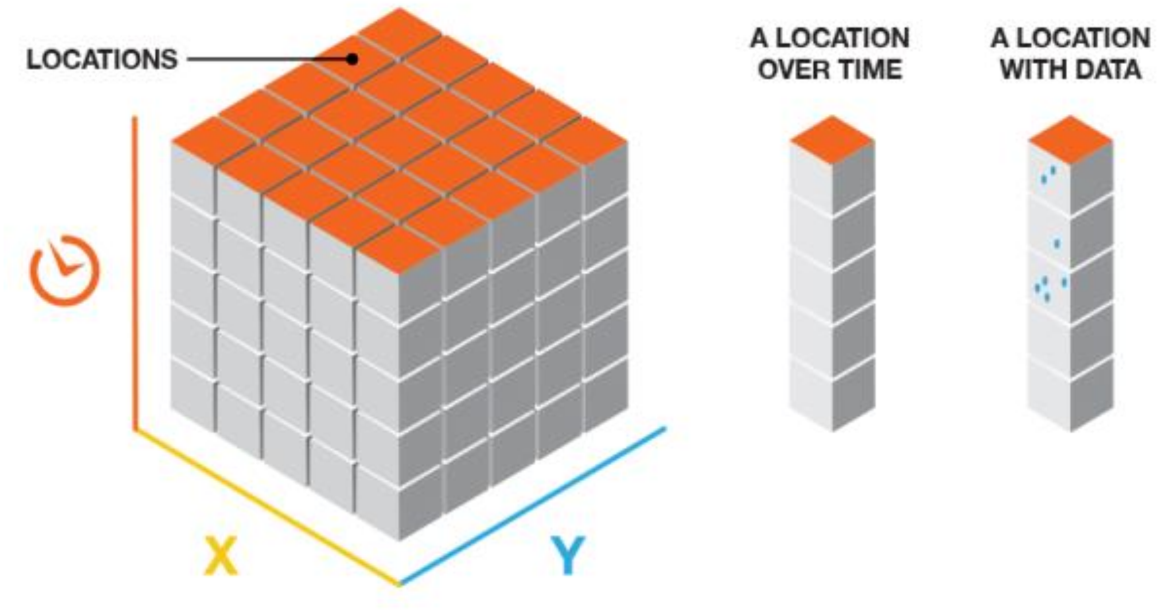


Figure 1 Abstract visualization of a temporal data cube (taken from the ArcGis documentation).

1.3. Representation of the Data Set in a Data Cube

The datasets in data warehouses are functionally a representation of both a domain and co-domain and the function of which transforms the former in the latter. In a geospatial representation, the “coverage” domain (space and time recorded) is mapped into a “coverage” co-domain (the set of parameters of interest to the goal). Theoretically and in a global application, both the domain and codomain can in and of itself be multi-dimensional, with set values (integer, real, complex numbers).

As an example, when taking data from EO satellites, if we take a set of observation from two different instruments (i), in two different timeframes (t), and focus on the soil-temperature (ST) and pressure (P), “I” and “t” would be a pair of coordinates forming a domain, and “ST” and “P” would be a pair of coordinates forming a co-domain, directly dependent on the other two. This could be represented by a function such as:

$$f: I \times T \rightarrow P \times ST \qquad (1)$$

Alternatively, it is noted that generally, the parameters can be moved from the domain to the co-domain (not the other way around unless its relation its bijective). In this case, assuming we have n measurements from an instrument/time pair (Say, Satellite X on the day Y), we can represent that measurement (in range of n) by a tuple of Pressure, Soil Temperature, Instrument and Time. This hypercube could now be represented by the following function:

$$f''[1, n] \text{ -> } P \text{ x } ST \text{ x } I \text{ x } T \qquad (2)$$

1.4. The major concern and EO data cubes principles.

Data Cubes should apply what is formally known as **dimension neutrality**, which means that the query made to the dataset and its complexity, should in no way be dependent on which dimensions are involved in said query. In the second hypercube described above, the dimension neutrality is in theory maintained, with all the parameters being treated the same. Asides from this major point, when designing a Data Cube in the EO domain, there are four critical principles that should be considered [10].

1) The Data-Cube must act as a digital system to manage data hypercubes and data analytics of said hypercubes.

2) Data-Cube must be able to manage N-dimensional hypercubes, and one of said dimensions must imperatively represent a geospatial location.

3) The Data-Cube system must be easily accessible and interoperable at a programmatically level, on the front-end of the system.

4) As with any digital system, there must be a governance mechanism.

1.5. Data cubes of Earth observation data for ecological monitoring

With the recognition of data cubes as valid and useful data structures, we saw a development of a specific type of data cubes as a response to the Earth Observation demands for Satellite Imaging, known as Spatial Data Cubes, holding 3 dimensions at his core concept, Longitude/Latitude/Time. As science and programming mechanism developed, at an institutional and public level, a 4th dimension was introduced and generalized, representing any given set of spectral layers present in a dataset of satellite imaging.[11]

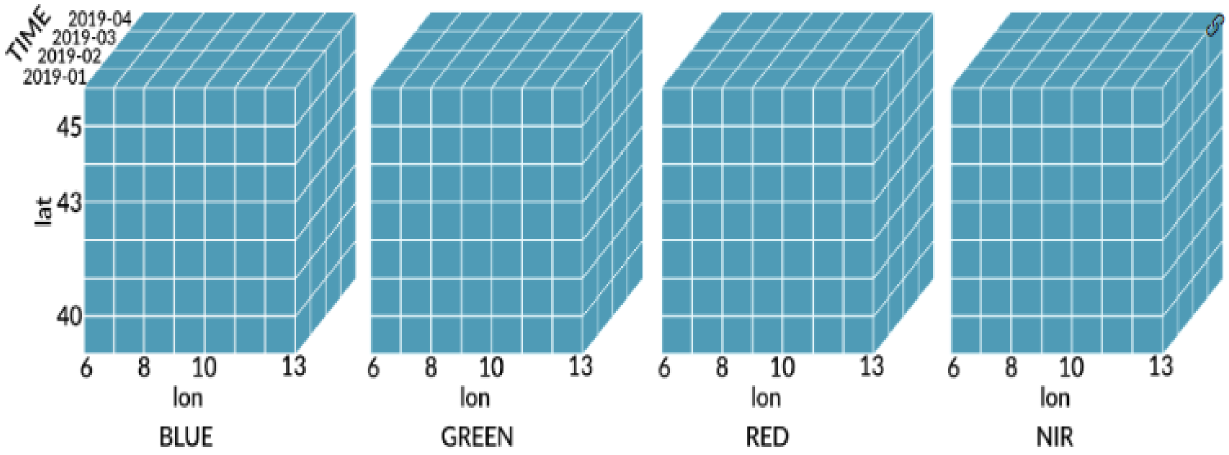


Figure 2 Four different multispectral geospatial data cubes

1.6. From Global to National Data Cubes

On a global level, the Open Data Cube initiative was founded and supported by the Committee on Earth Observation Satellites (CEOS) as a program meant to tackle the need for spread out availability of Analysis Ready Earth Observation Data.

Since then, it has released an open-source toolkit which helps develop analysis ready data and a subsequently derived data cube. Ever since, four data cubes are fully operational at a national level, in Colombia, Switzerland, Taiwan, and Australia (the first data cube on a national scale). In addition to those, in 2018 the African Regional Data Cube (ARDC) was launched, based in Nairobi, covering Kenya, Tanzania, Sierra Leone, Ghana and Senegal.

The ODC expects for 22 more national data cubes to be fully operational in 2022. Asides from the ODC, since then the European Union has funded the EarthServer project, which also deals with remote sensing data, using the RASMADAN technology for array databases[12]. In Brazil, another technique, the SciDB array method, has helped develop a data-cube used to produce land classification over large areas[13].

1.7. Australian Data Cube

The AGDC (Australian GeoScience Data Cube) was developed across multiple years with the mindset of using the Land Surface Imaging available since the first participation by Australia in the Landsat Program in 1979. The AGDC is set upon a “digital earth” view, composed of Earth Observations obtained by mostly remote sensing, with the support of land sensors and calibrations, duly processed and stored in a high-performance computing, allowing for the end-user to use that ARD in order to monitor, study and project the state of Planet Earth[6].

This was at the time the first fully operational EO National Data Cube and so it has become the staple on how to obtain and process the data and create a final product easily accessible to the end-user. It introduced the concept of large-scale geospatial data cubes as a valid and tested data structures for Earth Observation at a national level.

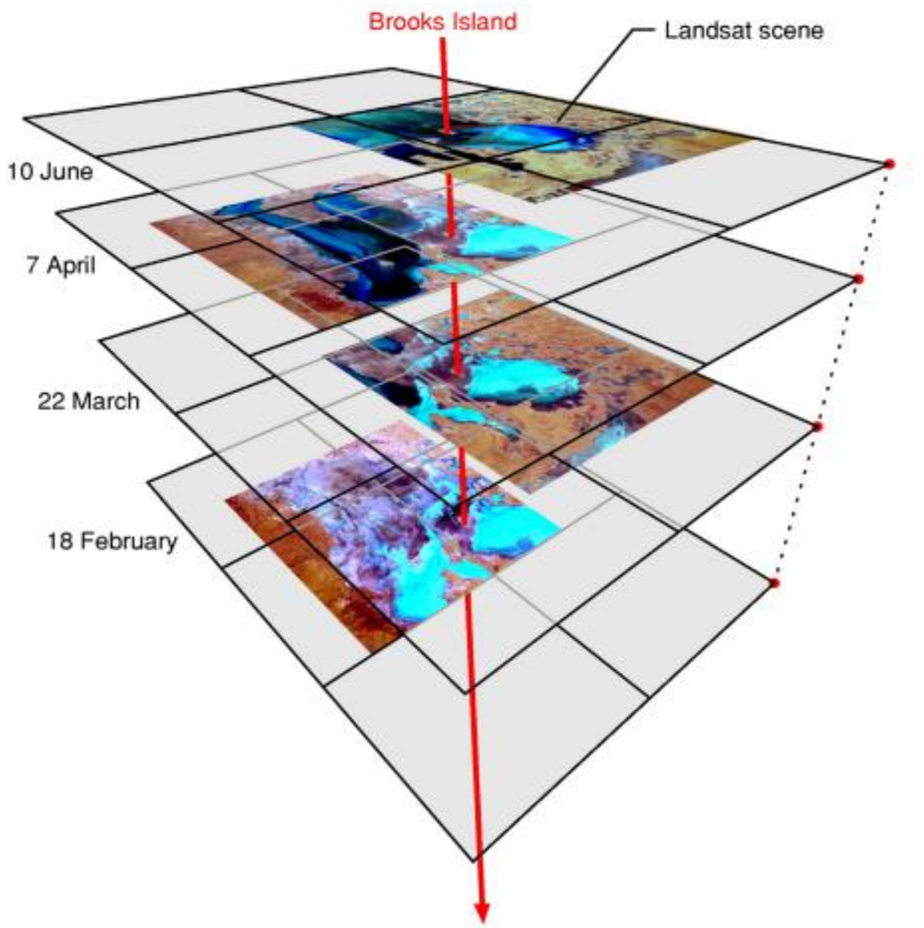


Figure 3 Idealization of the Australian GeoSpatial Datacube [6]

1.8. Swiss Data Cube

The Swiss Data Cube has since been founded, developed, and maintained by the GRID-Geneva. The SDC serves as support to the Swiss government for monitoring/reporting, and for Swiss institutions to use it as a motor to a better knowledge and study of the surrounding environment [14].

The SDC faced a major problem in its development, the general lack of efficiency in the pipeline Ancillary Data to Final Product. This, helped by the multiple sources of data, posted a necessity to create a unified/standardized procedure to obtain the data, process it to an ARD point, and subsequently deposit it the Data Cube.

As the Swiss landmass was the only point of interest for the SDC, a Python script was written to obtain the scene ID's available in the repositories (GEE, AWS, USGS) for a predetermined coverage area. Between 1984 and 2017, the total amounted to 3368 scenes, with a total size 867.5 GB of data. The SDC is based on the LiMES framework, which massively improved the efficiency of the whole pipeline.

Temporally speaking, what were hours of manual requests, the LiMES framework transformed into an automized pipeline of discovering the scenes, downloading, pre-process them to an ARD threshold, index them and finally ingest them into the Data Cube in the span of four minutes per scene. As of the testing phase, finished in 2016, the whole process was being ran in a infrastructure composed of: Processors Intel Xeon E5-2660 v2 @ 2.2 GHz; 8 CPUs (6CPUs used for processing, 2CPUs for system and UI); 50 Gb RAM; 2 TB Hard Drive; Linux Ubuntu 16.04 [14].

On the processing stage, the six processes are executed in parallel, while the indexation and ingestion phase must not be running separately. To facilitate this, the scenes were split into groups and inputted. The whole dataset of scenes was processed, indexed, and ingested in a 9-day span.

1.9. Environmental monitoring and remote sensing

The advancements in Data Cubes projects and technologies brought a new realm of possibilities for the use of geospatial data. The larger processing capacity and storage abilities allowed us to gather information at a broader and deeper scale, as well as introducing concepts of predictive models[15].

Even though the output of the data cubes might be the bigger role in this symbiose relation, there is also a side of input by these programs, which provide in-situ data not available from the EO and remote sensing techniques, while also serving as calibration mechanisms for them.

EO data cubes can be used to analyze and monitor environmental spatio-temporal patterns, namely changes and their impacts on biodiversity at national and regional scales. In particular, we wish to explore how sensed EO data, together with advanced data analysis and modeling ed to support biodiversity modeling pipelines fed by structural (e.g. land cover, vegetation extension) and functional (e.g. productivity, water balance, soil temperature) spatiotemporal datasets.

Group on Earth Observation (GEO) is global network interconnecting governments, academic and research institutions, data providers and businesses. The global collaboration and communication helped identify gaps and reduce duplication of efforts in the areas of sustainable development and sound environmental management, leading to a bigger progress in the areas. The crown jewel of the GEO consortium is the Global Earth Observation System of Systems (GEOSS), which aims to integrate observing systems and obtained data to a better efficiency, inter-operability and accessibility.

One of the main reasons behind the GEOSS formation are of course the financial benefits behind a better understanding of our planet, and better projections of how it may act. One simple example are the droughts, which could be better predicted with a better understanding of the impact of agriculture output on the soil, and that in the US alone produces around 6 to 8 billion dollars of damage[16].

The entire new industry/academic field is supported by the huge amount of growth in Satellite Imaging in both spatio-temporal, radiometric and spectral resolutions, volume of data and availability since the turn of the millennia. In 1999, with the launch of Terra, NASA propelled the first satellite-based observation system of Earth and its inherent processes. While minor personal satellites are also widespread, between NASA and ESA alone there are around forty different satellites continuously providing data, from the Sentinel and Landsat Missions (https://www.copernicus.eu/en/about-copernicus). Maximizing the use and cross-reference of data of all the available sources is a goal and requirement of any project in the study field.

1.10. Thesis Aims, Challenges and Motivation

As the social and environmental pressure exacerbated on country increase, having a comprehensive understanding of the biodiversity and land composing our country, as well as its interconnection and relations is a priority and of critical interest [5].

With Satellite Imaging becoming widespread and correctly, the current rate of growth in the availability of satellite data is only bound to increase, with the CEOS reporting an increase from twelve in 1980 to over sixty-nine operational EOS missions in 2014, to one hundred and ninety-seven operational EOS missions as of 2022 [17]. While an accurate prediction of how large the data availability will be on a national level long term is a tough task at this day and age, we must prepare ourselves for the inevitable need for bigger and more efficient pipelines of work to deal with it. It is also expected that for any amount of raw data obtained from satellite imaging, the processed data derived from it would lead to a 3 to 5-fold increase in needed storage space, consistent with the work done by Overpeck - et al. [18].

To tackle this, we aim on a larger scale to create a verified, assessed and evaluated work pipeline for obtaining, processing, storage and finally make satellite data available to the public. With this larger scope and aim in mind, we decided to explore the creation of a data cube including data from the entire continental portion of Portugal. One of the big motivations behind our work was also making sure the entirety of the process, from the images selected to the processing stage was made with the aim of full open source and free availability, which was considered in multiple steps of our work.

As a proof-of-concept of this broader scale national data cube and the objectives behind it, we created a smaller scale data cube and front-end visual interface hosted on a permanent 24-hour cycle, also creating a set of predetermined exercises of his utility, such as spectral indexes computation across diminished timelines, variable statistics across space and/or time spans and the ability to correlate those to unlikely or special events, in this care forest fires.

We also compare it to already used workflows using other methods of data stacking (namely *rasterStacks*) to also cross check the processing speeds and memory usage of both workflows to help consider the hardware needs of the final product and data cube.

The entirety of the project (as well as the support network behind it) was designed prepared and made under the scope of the SeverusPT project (https://severus.pt/) with direct links to the PorBiota project (https://www.porbiota.pt/).

2. Methods

2.1. Overview of the methodology/workflow

In this section we will go over the 2 main phases of construction of the data cube, the pipeline that handles the download of metadata and image data from the MODIS servers, and then a primarily local coding phase which proceeds to ingesting those same primary level data products into the data cube, as well as creating secondary level ones for the same purpose.

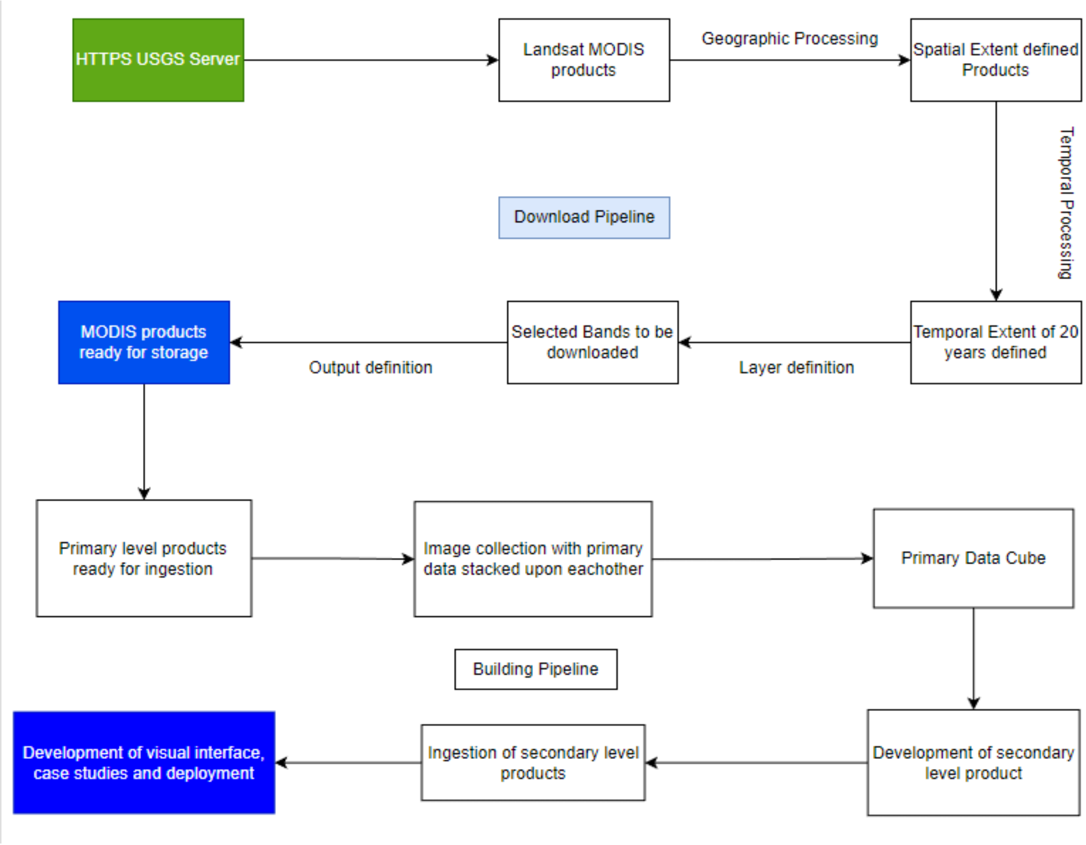

HTTPS USGS Server
Landsat MODIS products
Geographic Processing
Spatial Extent defined Products
Temporal Processing
Download Pipeline
MODIS products ready for storage
Output definition
Selected Bands to be downloaded
Layer definition
Temporal Extent of 20 years defined
Primary level products ready for ingestion
Image collection with primary data stacked upon eachother
Primary Data Cube
Building Pipeline
Development of visual interface, case studies and deployment
Ingestion of secondary level products
Development of secondary level product

Figure 4 Complete Data Cube building pipeline.

2.2. Image transfer

Satellite Imaging as of 2022 can be obtained from multiple core providers. The first distinction and choice to be made in the project and in a larger scope building and maintenance of a National Data Cube is which products we are aiming to maintain and process/ingest into the Cube. At this time and as model raw data for our proof-of-concept outputs, we decided to use three different LANDSAT products, MODIS11A2 (8-day composite of Land Surface Temperature values), MODIS09A1 (8-day composite of Land Surface Reflectance values) and MCD84A1 (MODIS product with monthly burned areas).

The download and pipeline were automated off a self-written R script based on the MODIStsp R package. This package was built and has been sustained since 2015, becoming the prime package for MODIS products downloads. Allowing for geometric, temporal, radiometric processing, it is the foundation of the first part of the pipeline towards this Data Cube, while being completely open source.

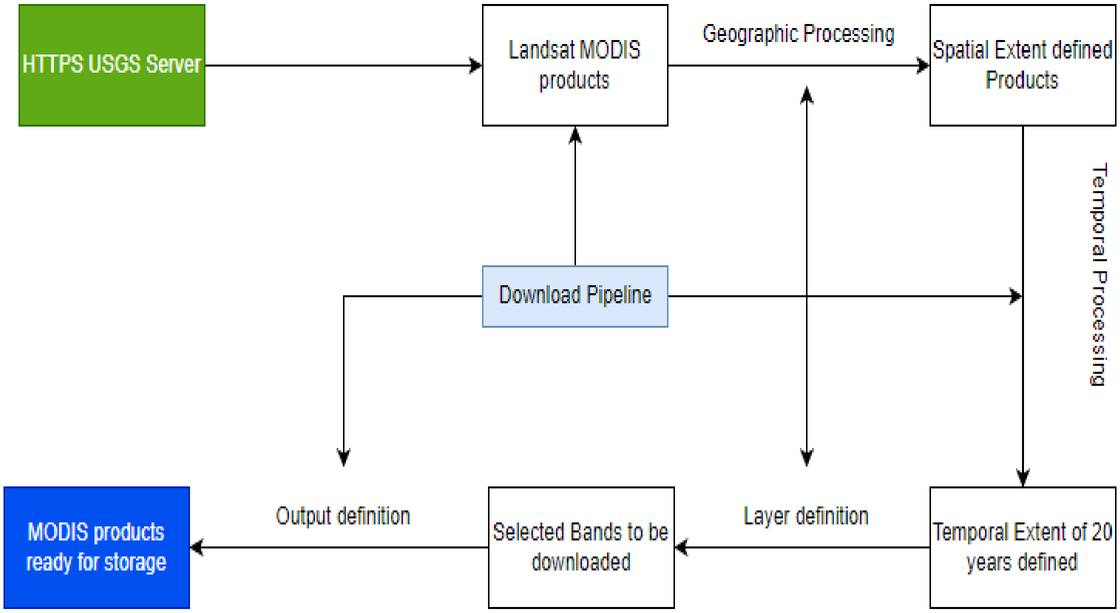

HTTPS USGS Server
Landsat MODIS products
Geographic Processing
Spatial Extent defined Products
Temporal Processing
Download Pipeline
MODIS products ready for storage
Output definition
Selected Bands to be downloaded
Layer definition
Temporal Extent of 20 years defined

Figure 5 Designed download pipeline for any MODIS product.

While lengthier and more straight forward ways of using their GUI are available to the user if needed, we developed three distinct .json files that have written the entirety of the options available within the package, to obtain the desired products ready for storage. These files contain geometric, temporal, and radiometric/band-selection processing options.

Geometrically speaking, there is a defined option for the Spatial Extent, which we defined as a bounding box of the Portuguese continental extent, the output projection of the output Geo Tiff files, which we specified by using the standardized ESPG Portuguese code, 3763, and resampling the pixel size to 500 m per pixel, using the nearest neighbor's method. This resampling was literature and logic based, as other known and usable resampling methods, like cubic or linear, are not usable on quality-based variable (Pixel Quality Layers are involved in our products) and aren't recommended on continuous variables such as surface reflectance, causing contamination of high-quality pixels with values from low-quality pixels (Figure 3).

As far as Radiometrically processing options, we can select which of the available bands in each product we want to store, which may help with the partitioning of storage space in possibly less than ideal hardware components for future use of smaller end users of this pipeline/workflow.

2.3. Temporal extension and resolution

As far as temporal definition goes, there were two different scopes to be dealt with. At a first level, for the current project and initial version of the prototype, we decided to use a set length of 20 years for the three products in question. This takes the data structure from January 1st 2001, to January 1st 2021. This encapsulates enough of a timespan that it allows for the initial creation of sizable time-series as required, with enough base data for statistical relevance.

Subsequently, there was a need for some sort of real-time API possibility. In order to achieve this, we used both the MODIStsp package and the TaskScheduleR package, a R based packaged which uses administrator authorization and privileges to the Windows Task Scheduler properties to create set scheduled tasks of R scripts. With this possibility and framework in our mind, we created a R script which iterates over the 3 different ingestion files previously referred with the *end_date* argument set as the current system date. This will make it so whenever the scheduled task is run, the underlying MODIStsp package will search for the newly available products not yet downloaded (table1).

As it currently stands, the script and scheduled task script are defined to run on a weekly basis, meaning every Sunday the system would run both, obtaining all (if available) new scenes for the defined hyperparameters of the pipeline (geometrical and radiometric). This second set of scripts is to be used in addition to the User Interface prototype referred to in a posterior phase of this work. As a note, while we set the previous end-date at a future date so that the script may function as a real-time API with success (until the 2030 date), efforts were and are being made so that the options file associated with the scheduled task script is passed a sys.date() argument, which returns the current date and time of the OS(Operating System) reading a code snippet. Unfortunately, as of this time, and with JSON as an extension that acts only as a transportation format, we cannot pass arguments with logic inherently behind it, such as the sys.date() one.

```
opts_files <- c(file.path("C:\\Users\\nunoe\\OneDrive\\Ambiente de
Trabalho\\MODIS09A1\\MODIStsp_MODIS09A1.json"),
                file.path("C:\\Users\\nunoe\\OneDrive\\Ambiente de
Trabalho\\MODIS11A2\\MODIStsp_MODIS11A2.json"),
                file.path("C:\\Users\\nunoe\\OneDrive\\Ambiente de
Trabalho\\MODIS14A2\\MODIStsp_MODIS14A2.json"))

for (opts_file in opts_files) {
  MODIStsp(gui = FALSE, opts_file = opts_file, verbose = TRUE, parallel = TRUE)
}
# MODIS09A1 outputs
out_fold <- file.path("C:\Users\nunoe\OneDrive\Ambiente de Trabalho\Outputs\MODIS09A1")

# MOD11A2 outputs
out_fold <- file.path("C:\Users\nunoe\OneDrive\Ambiente de Trabalho\Outputs\MODIS11A2")

#MCD64A1 outputs
out_fold <- file.path("C:\Users\nunoe\OneDrive\Ambiente de Trabalho\Outputs\MODIS14A2")
```

Table 1 Code loop of the download pipeline

Alternatively, and since HTTPS and credential secured servers inherently produce some errors on some scripts, especially if authorizations requirements are updated and the R package in use isn’t updated/stops being maintained, we produced an alternative download script for the MODIS products. This script works as a shell command line, which runs either on the base command line in LINUX or in a command line simulator in Windows. In this work we used Cygwin, a Windows focused GNU and OpenSource functionality emulator, allowing for a Linux based shell approach. The scripts were written for the 3 products. An attempt on building a pseudo-API approach as provided for the R script version was also developed for further publishing.

In any of the ways, the output .HDF files obtained from the LP DAAC Data Pool Landsat database are always processed under MODIStsp, either simultaneously with the download, in the R script model, or at a local level post-download, with the Shell Script model. Both produce the desired products ready for data cube ingestion. The Shell Script is provided in the attachments section of the dissertation.

2.4. Data Cube building

As we went over in previous sections of this work, we aim to build a n-dimensional data structure known as Data Cube. On a first note, while there are multiple Data Cubes providers and developers, most functioning large scale data cubes and correspondent pipelines are maintained in HPCs, high performance computing systems, either with the use of super computers or the use of high performing clusters and grids. While we do believe that at some point in the future our base project can and should be moved to a similar hardware structure, both the proof-of-concept data cubes and image downloading were made on more modest desktops and servers.

As such, and after review, we built our data cubes using a set of personally written scripts using the gdalcubes R package as the basis for our work. This ensured that the entire pipeline of our work, from downloading, processing, data cube building, data ingestion, analysis and UI building were all made under a single programming language, providing a simple and clear-cut stream of syntax and logical code, easily interchangeable and adapted to different needs and aims, while also using peer-reviewed packages and code[11], [19].

The Data Cube building process consists of several consequent steps which we will demonstrate using the MODIS09A1 product, the 8-day composite of Land Surface Reflectance values. This product consists of pixel values for eight different spectral bands with each pixel representing 500 m area, as well as a Quality Assurance layer. For every sequential eight days, the highest quality value for each of the pixel is chosen amongst the eight-day sample. The original provided MODIS products are also already corrected to compensate for atmospheric conditions, such as aerosols and gasses[20].

```
Files = list.files(paste(wdir,”MODIS09A1”,sep=“”), pattern = ".hdf", recursive = TRUE,
full.names = TRUE)
Files1 = list.files(paste(wdir,”MODIS11A2”,sep=“”) pattern = ".hdf", recursive = TRUE,
full.names = TRUE)
Files2 = list.files(paste(wdir,”MODIS11A2”,sep=“”) pattern = ".hdf", recursive = TRUE,
full.names = TRUE))
LSR.col <- create_image_collection(Files, "MODIS09A1", "MxD09A1.db")
LST.col <- create_image_collection(Files1, "MxD11A2", "MOD11A2.db")
Fire.col <- create_image_collection(Files2, "MxD14A2", "MOD14A2.db")
```

Table 2 Ingestion of the MODIS products towards the data cube

The gdalcubes package works with a first step of creating an image collection, a set of n images corresponding to a determined spatial extent, with each image containing m band values (table 2). In some data products like Sentinel mission ones, image data may come from different files. In MODIS products, the raw .HDF contains data for all layers. To create an image collection, we must pass an ingestion script to the package, which tells the package source code how to read both the file folder and the file names and the file itself.

After making sure the package has the correct product raw files folder and extension type, we need to make sure it can read the files. While the base package has the ingestion files for some of the available MODIS products, we also wrote from scratch the ingestion file for a few of our desired ones, MODIS09A1 included. The code writing for both this and the MODIStsp option files referred to in the previous section was made using IntelliJ, a JavaScript IDE with a .JSON plugin, and both are annexed in the final section of the dissertation.

Using the source book on the MODIS product, which gives us the band-names, the regular expressions needed to read the filename in the folders, to read them in a temporal timeline and assign dates to each as well as the fill values and no data values. The argument “pattern” refers to the name attributed to the bands/date and time in the raw .HDF file obtained from the download pipeline.

As of this step, we have the image collection with the band value for the date and time extent we required. The next phase involves the creation of the geometric delimitation of our data cube, which involves the spatio-temporal extent definition, the temporal and spatial extent of each cube cell (resolution), and the resampling methods when values from multiple images occupy the same cell in the cube. By default, we used the nearest neighbor's method for both resampling methods, product of the data volume in question, considering the processing times that would ensue. Specific resampling methods may prove the best option depending on the final purpose of the user of the data cube. In this case we used a bounding box to describe an approximation of Portugal, including all of Portugal's landmass, projected with standardized ESPG code for Portugal[21],defined the idealized one-year timespan and set a spatial resolution of 5km per pixel (table3).

```
V.Portugal = cube_view(srs="EPSG:3763", extent=list(left = -121000, right= 164000,
bottom = -301000, top = 278000,  t0 = "2016-06-01", t1 = "2020-01-01"), dx=500, dy =
500, dt="P8D")
```

Table 3 Geographical, temporal and resolution definition of the data cube.

We can now head onto two different paths, depending on the purpose behind it. Gdalcubes, as platforms like Google Earth Engine, uses a "lazy" approach to the data cubes, with the users making their own analysis and inputs and only after said inputs is the processing done, while all the way up to the final input, proxy objects are created, maximizing the memory usage and processing times. Instead of recreating a cube with every parameter passed, only the final function call, usually plot() or animate() , returns a complete object.

2.5. Complete process and notes for further interoperability

As such, if we discuss the use of this package and code structure on a national data cube, if we can assume and assure the use of an appropriately built base server, the optimal setup is to create and merge (code for this already made and set up) the data cubes for each MODIS Product, creating one final structure containing the layers of each MODIS product for the spatial and temporal extent defined. At that point, the only final requirement is for the analysis to be input upon said cube, where the code would run the necessary queries and return the final objects. If we aim to just create the Data Cube itself with intention to use it on another pipeline, we can choose to simply write it as a .NetCDF file, a self-describing, machine-independent data format that supports the creation, access, and sharing of array-oriented scientific data. This is the international standard for the Open

GeoSpatial Consortium. The major benefit stems from its exportability towards other languages, with CDF formats being already readable and accessible/writable in classical languages as C, C++, Python and R, as well as newer more advanced languages as Pearl, Ruby and Octave.

```
library(gdalcubes)
library(dplyr)
#See base package supported sattelite products
collection_formats()
#Adding the created ingestion .json file to the package to allow for MODIS09A1 product
ingestion
add_collection_format("https://raw.githubusercontent.com/NunoFilipeSousa/Thesis/main/MO
DIS09A1.json", name = NULL)

#Path to the file lists and making sure it only reads the .hdf files
Files = list.files( paste(wdir,”MODIS09A1”,sep=“”) pattern = ".hdf", recursive = TRUE,
full.names = TRUE)
#creating the image collection and delimitate cube
M.col = create_image_collection(Files, format = "MODIS09A1", "MxD09A1.db")
v = cube_view(srs="EPSG:3763", extent=M.col, dx=5000, dt="P8D", aggregation = "mean",
resampling = "average")
Final_MODIS09A1_Cube = raster_cube(M.col, v, mask=image_mask(band = "QC_500m", bits =
0:1, values=c(0,1), invert = TRUE))
#write out the data cube in a cross-language format
write_ncdf(Final_MODIS09A1_Cube)
```

Table 4 Complete pipeline of creation of a data cube.

As such, and regarding the base structure of the main Data Cube, we go from creating an image collection with the metadata, to creating the geometric delineation of the data cube pretended to merging both, creating the super-intended version of our end data cube. From here on now, we either run the queries on the base version data cube, or, if we know à-priori that we will run a single pipeline of work regarding only specific bands, one specific timeline or one specific spatial extent of the data, we can write out a specific data cube, to use in this same work pipeline, or in another format to be used on a different software(table 4).

For purposes of evaluation of method and comparison against other used workflows and work pipelines, we created three smaller data cubes, with the same 3 MODIS products and a one-year length using the same satellite tile extent. This allowed for processing speed evaluation, memory allocation evaluation and scalability purpose evaluation of the workflow. The code for both versions (the smaller evaluation cubes and the first stage prototype of the Portuguese Data Cube) is completed, with the latter ready for deployment, pending approval and delineation of the overarching coordination, organization, and management questions.

2.6. Performance Evaluation Methods

As with any new process or pipeline for data management and processing, it is important to take notice of its computing performance. Unfortunately, R as a language does not have immense third-party support for performance testing, neither on the form of packages designed for this purpose, nor in the academical form, with guidelines or papers over the subject. Instead, we chose to evaluate the process over a more streamlined performance test, applicable and studied on a general setting, not specific to R[22].

As such, we focused on the CPU usage and impact and the time performance of the scripts, as both at a local level, it is the most important variable to measure, and in an eventuality of use of cloud-services, said time performance also directly impact cost of server rental/allocation. We focused on evaluating the scalability and stress performance testing of the pipeline, as this aspect of the whole project and Data Cube is the one predicted to be strained the furthest as it develops.

For a better understanding of the capabilities of the code, we measured the processing speeds of the queries, and then also evaluated the impact of the resolution of the data cube pixels. RAM Memory usage in R is inherently locked to the max available memory usage of the PC in the x64 bits version, and as such we refrained from altering such variable with any kind of third-party software. For all purposes of this process, we had 16 Gigabytes available RAM memory, at 3200 Mhz speed. At no point during the process of requesting queries or building the data cube did the R process use over 800 MB of RAM memory. As far as CPU usage goes, while the code can be defined to run on any number of CPU cores, core numbers < 4 led to instability of the process, resulting in multiple crashes of the R sessions during the performance tests. As most currently commercially available machines, from the low-end laptops to higher end servers all present at the very least 4 and in most cases at least 8 cores, we decided to leave CPU influence from a side as a complete evaluation. Preliminary tests and literature showed a logical increase in performance the higher the number of cores, but there was a threshold from which the performance increased became negligible when dealing with higher spatial resolutions [11].

3. Results and discussion

3.1. Resulting Data Cube

The resulting data cube is a multidimensional data cube with an overseeing geographical delimitation of the continental area of Portugal. The structure built for the purposes of the dissertation encapsulates 10 years of data, from January 1st 2010 to January 1st 2020.

In more detail, other than the 3 previously referred "axis" of the cube, there are 13 other spectral dimensions currently included in the structure. From a primary product standpoint, there are 10 spectral bands directly obtained from the MODIS09A1, MODIS11A2 and MODIS14A2 products. These are the 7 reflectance bands, Red, Blue, Green, Near Infrared and 3 Short-Wave Infrared. In addition to those, there are also 2 Land Surface Temperature bands, for daytime and night-time, as well as a Fire Mask band, used for fire detection and delimitation. We also computed and added as secondary products 3 spectral indexes, NBR (**Normalized Burned Ratio**), NDVI (**Normalized Difference Vegetation Index)** and SAVI (**Soil-Adjusted Vegetation Index)**.

From a granularity standpoint, given that the 3 products have different spatial resolutions amongst themselves, with the MODIS09A1 product being a 500 m x 500 m product while the other two are 1 km x 1 km products, we used the highest common resolution, and hence the cube has a 1 km x 1 km resolution. Nonetheless, if needed, the pipeline is set up to use an even higher spatial resolution, if assured that the only used products are the MODIS09A1 derived observations, as of now. All of the observations in a data cube follow an 8-day composite norm, meaning that for an X Day value, that value represents the best value taken in a group of 8 days of observations. This was done to ensure the least amount of impact was done by atmospheric/aerosol/cloud disturbances.

As currently constructed, the data cube includes images and data amounting to 70.1 Gigabytes of data.

3.2. Monitoring and Environmental relevant queries

To make a demonstration of the possibilities behind the data cube, we designed a series of consequential queries to achieve a certain final product to be downloaded as a time-series, visual information or supporting data for further programming interactions, with special attention to machine learning. Different end-product needs have different coding paths and associated costs, whether time, difficulty, or data accessibility to the average user.

First up, we computed a series of plots visualizing the spatial distribution for values of two assorted products, the LST (**Land Surface Temperature**), the NVDI (**normalized difference vegetation index**) . The first one is a straightforward measure of the Earth's Surface Temperature at a set spatial polygon, measured by thermal reflection[23]. This is obtained by getting the pixel value for all pixels inside our delineated geometric form, for the date-spans we choose. We selected 3 different years, 2010,2015,2019 and plotted the mean LST value for the 30 days of May and November, representing season change. We used 1KM as the spatial resolution, since LST measurements, made aboard the satellites are capped at 1km spatial resolution in the MODIS imagery

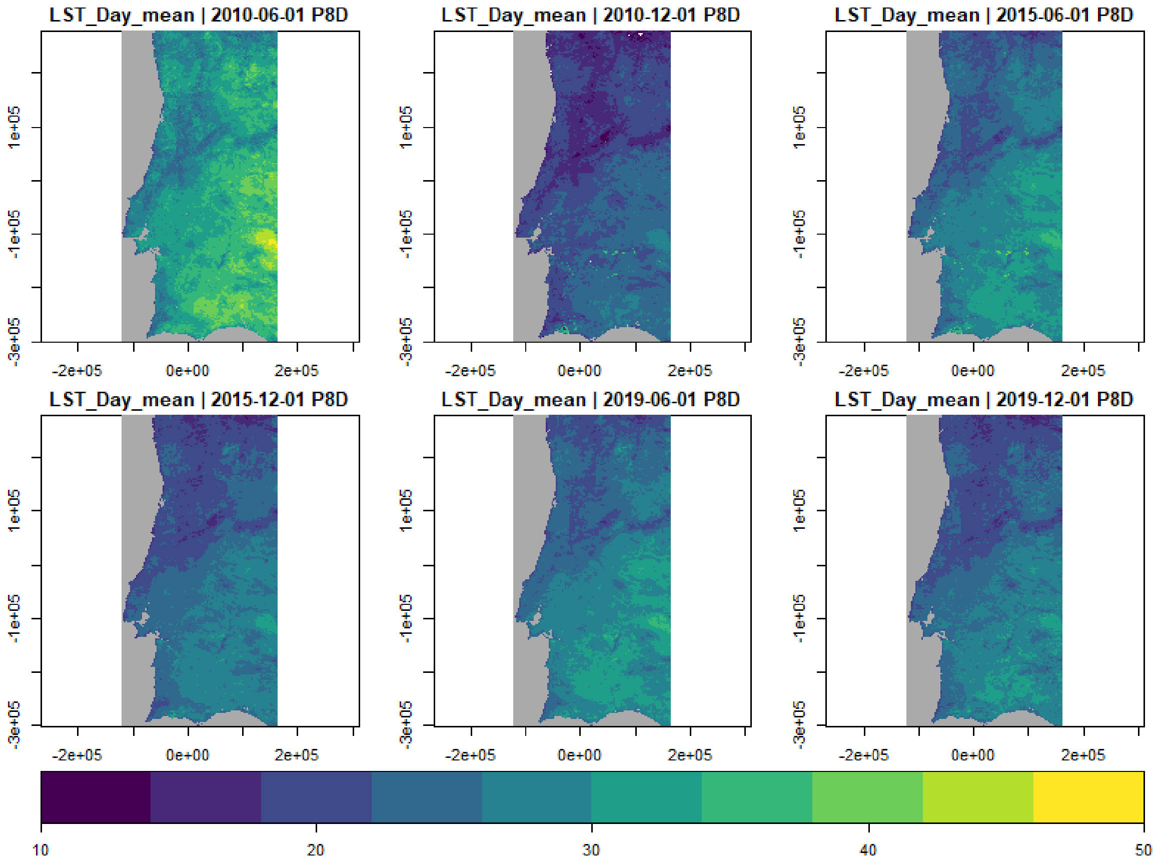

LST_Day_mean | 2010-06-01 P8D
LST_Day_mean | 2010-12-01 P8D
LST_Day_mean | 2015-06-01 P8D
LST_Day_mean | 2015-12-01 P8D
LST_Day_mean | 2019-06-01 P8D
LST_Day_mean | 2019-12-01 P8D
1e+05
-1e+05
-3e+05
-2e+05
0e+00
2e+05
10
20
30
40
50

Figure 6 Mean LST values for the month of May and November, for 2010,2015 and 2019

While the winter season changes are present albeit scarce and further data management options would then ensue (as done with a posterior case study), the Land Surface Temperate values from the summer of 2010 to 2015 are eerily visualizable in the current visualization form. LST can reducibly be labeled as a measurement of the Earth's land surface temperature, making it a prime indicator for the energy partitioning across the planet's surface, having been classified by the International Geosphere and Biosphere Program (IGBP) has one of the most important spectral measurements to study and explore [24]. It has been used in multiple environmental studies, from deforestation impact on LST [25] to an assessment of heat waves impact on land surface measurements, a work which has related severe heatwaves to drastic changes in an area's LST values[26].

Most studies around impact on LST and what LST may or may not represent use its max or mean values pre and post target date/occasion, and as such, we figured the first step would start by a visualization of the change of LST over time, such as in figure 6.

```
LSTVALUES = select_bands(Merged_Portugal_Cube, "LST_DAY") %>%
    apply_pixel("LST_DAY * 0.02 - 273.15", names="LST_Day") %>%
    select_time(c("2010-06-01", "2010-12-01", "2015-06-01", "2015-12-01", "2019-06-
01", "2019-12-01")) %>%
    window_time(expr = "mean(LST_DAY)", window = c(30,0)) %>%
    plot(key.pos = 1, zlim=c(10, 50), col=viridis::viridis),
    animate(key.pos = 1, zlim=c(10, 50), col=viridis::viridis, fps = 5) #gif
encoding instead of a layout of plots
```

Table 5 Code to obtain the plots in figures 6, the LST values. Animation as gif was included but cannot be visualized in word document

For an easier visualization of the changes through time, to detect timeframes of interest for further investigation, we did create an alternative in which the consecutive plots are encoded into a .gif for easy information retrieval. Unfortunately, there is no way to provide said file in a .docx document, but we did provide the alternative line in the code above on table 5, which takes us through the process to create the figure above, in which the temporal subset can be taken as simple date lists or a temporal subset, which would create a visualization in temporal order.

Since this is a single-band plot, from here on now we can use the function *extract.geom()* to get the values for shorter time frames and specific sf objects, polygon-based objects representing specific areas. We go through the entire process in the section “Pedrogão as a case study” detailing the entire pipeline from cube building to value extraction to information building.

In the second query, we made the same request than before but applied it to two different spectral indices. NDVI, formulated by the equation $NDVI = \frac{\mathrm{NIR} - \mathrm{Red}}{\mathrm{NIR} + \mathrm{Red}}$, where NIR represents the Near-Infrared (841 to 867 nm) and Red the visible red (620 to 670 nm) channel of remote sensing imagery. This index has had a widespread use due to its easy visualization nature, allowing for a quick delineation of vegetation and vegetative stress for monitoring and study, making it appealable to commercial and agriculture studies[27].

We plotted both layouts in sequence (Figure 7), applying the same temporal selection as in the LST plot, to detect and visualize delineation of vegetation-heavy zones and possible tendencies which may have showed up in 10 years, having in mind possible future work.

```
NDVI=select_bands(Merged_Portugal_cube, c("B01","B02")) %>%
    apply_pixel("(B02-B01)/(B02+B01)", names = "NDVI") %>%
    select_time(c("2010-06-01", "2010-12-01", "2015-06-01", "2015-12-01", "2019-
06-01", "2019-12-01")) %>%
    filter_pixel("NDVI > 0") %>%
    window_time(expr = "mean(NDVI_30D)", window = c(30,0)) %>%
    plot(key.pos=1, zlim=c(-0.5, 1.0), col=viridis::viridis)
```

Table 6 Same code format as the table before, this time produces the image in figure 7, the mean NDVI values for the months of May and November in selected year.

The code on table 6, and the one before, follow a linear progression in which the original data cubes created from the first lines of the case are cached, allowing for the apply_pixel(), filter_pixel() and window_time() functions to be changed and ran again, saving the user time and the program memory usage. This option was only turned off when measuring the performance times of the scripts, as the cached data cubes reduced drastically the time of every repetition past the first evaluation since it never actually ran the queries again.

Speaking from a logical sense, the script is simple. We take the original Portuguese Data Cube with the 3 products, then proceed to select the corresponding bands needed for the final index we aim to obtain. B01, B02 represent red and near infrared channels of the MODIS data. The temporal selection can be made after by providing a list of the target dates, or if we wish to use a timespan, then crop() allows to subset the time variable by a t0 and t1 argument, after which the program will plot/obtain every value available between both. After having the required bands, we apply the arithmetic function required to obtain he desired index. This specific function takes regular expressions such as mean, min, max, standard deviation, but also user input expressions or functions. After this, we can apply a window_time() function, to obtain the values correspondent, in this case, to the 30 days before the given date. This allows for a more robust evaluation, as single day values can be either outliers or malfunction of the instrument, amongst others. This problem is also solved using 8-day composite products, which take the best quality pixel of the eight previous days, taking care of any possible atmospheric/cloud interference without the need for multiple quality masks for each day[28].

From here on now, it is a matter of graphical choices, the number of plots per column and row of the image, the color palette and choosing between plotting and animating, context dependent.

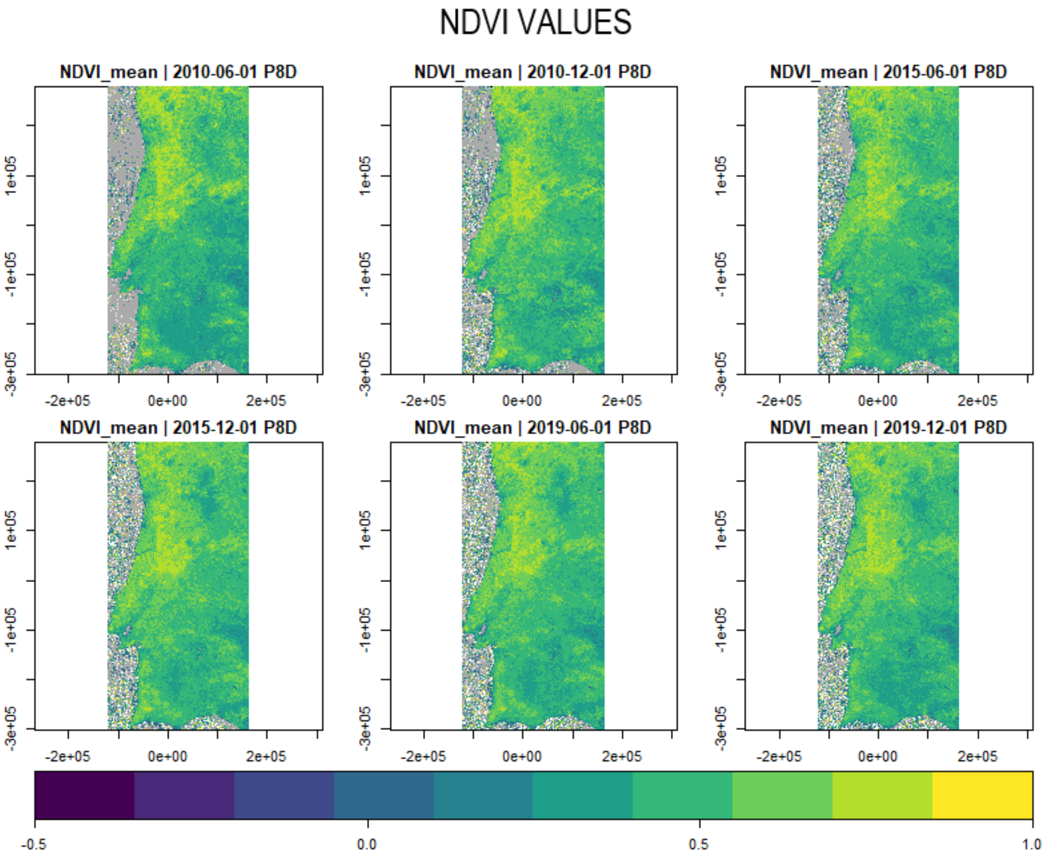

NDVI VALUES
NDVI_mean | 2010-06-01 P8D
NDVI_mean | 2010-12-01 P8D
NDVI_mean | 2015-06-01 P8D
NDVI_mean | 2015-12-01 P8D
NDVI_mean | 2019-06-01 P8D
NDVI_mean | 2019-12-01 P8D
1e+05
-1e+05
-3e+05
-2e+05
0e+00
2e+05
-0.5
0.0
0.5
1.0

Figure 7 Mean NDVI values for the months of May and November in 2010,2015 and 2019

We can quite easily see the delineation of vegetation heavy areas, while identifying the green vegetation free region in Alentejo, a more deserted area. Slightly more subtle but still noticeable is the lower average values for both in pixels across the Porto area. Ever since 2010, there has been a considerable urbanization of the entire district as the city center became more crowded, which may be an explanation for the loss of value in these indexes.

3.3. Case study as a display of possibilities

During the 10 years of data that we ingested in the cube created as a prototype for display and use during the dissertation project, one of the most important and impactful environmental occurrences to happen in Portuguese soil was the Pedrogão Grande fires in June of 2017. These started on the 17th of June and ended up as the deadliest forest fires in Portugal history, as well as produced a distinguished impact in the area, at a structural and personal level[29]. With such thing in mind, we decided to try and visualize the impact it had on the area using the tools included in the data cube.

```
PedrogãoFoto <- select_bands(Merged_Portugal_cube, c("B01","B04", "B03")) %>%
   select_time("2017-06-21") %>%
   plot(rgb = 1:3, fps= 5, save_as = "Pedrogão_after.jpg"),
PedrogãoFoto <- select_bands(Merged_Portugal_cube, c("B01","B04", "B03")) %>%
   select_time("2017-06-05") %>%
   plot(rgb = 1:3, fps= 5, save_as = "Pedrogão_before.jpg")
```

Table 7 Creation of the images in figure 8, created from the red, blue and green bands of the satellite imaging

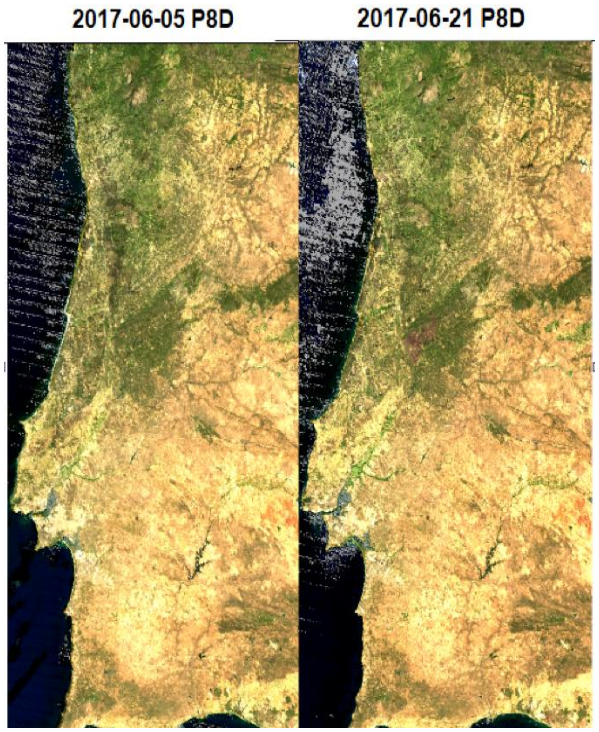


Figure 8 TrueColorRGB pictures obtained on the observation immediately before and after the Pedrogão Fire

First, we created two True Color RGB images from the MODIS data downloaded, one for the last instance before the fire and another one for the instance immediately after the start of the fires. We hoped, with success, that it would return a visual impact of what the fires did to the area. We omitted the creation of the data cube in the following images since the creation was unique and all following queries, from previous sections and this case study were all made upon the original cube.

As shown, there is a noticeable burned area showing up on the TrueColorRGB image plotted from the instance after the fire deflagration. Smoke and Cloud effects are inherently accounted for during the processing of the MODIS data, hence not showing up in the image.

From here onward, we started developing workable multi-band end-products from the dataset. Since we were aiming to obtain only the values for Pedrogão Grande itself, we used a shapefile of the municipality as the spatial definition from which to get the spectral indexes values. At a first stage, we made a distinct query for NDVI, and introduced two non-standard MODIS indexes, NBR and SAVI.

SAVI stands for Soil-Adjusted Vegetation Index, a spectral index developed from NDVI which was given a soil-adjustment factor L, to consider the vegetation density in the soil when obtaining the value [30] [31]. It is formulated by $1 + L\frac{NIR+Red}{NIR+R+L}$.

NBR stands for Normalized Burn Ratio, and it is used as a variable to both delineate fire affected areas, as well as classify the severity of the fire in distinct points of that area. It is formulated by $\frac{NIR-SWIR}{NIR+SWIR}$, with NIR being the Near Infrared Channel(841 to 867 nm) and the Short-Wave Infrared (2080 - 2350 nm) channels. For classification, we used the following delta NBR (NBR_Prefire_Values – NBR_PostFire_Values) to classify each pixel in the area regarding the fire severity. The classification was made according to thresholds designed

and used for the Mediterranean Area(Table 8). [32]

SEVERITY LEVEL	dNBR RANGE
Low Severity	<.319
Moderate Severity	.319 to .649
High Severity	> .649

Table 8 Severity level classification derived from DeltaNBR values

To work with this framework of data, we used the extract_geom() function mentioned before. This function allows us to extract the band values for a specific spatial extent. To define this extent, we downloaded a shapefile of Portugal, present in the CAOP2021 (Carta Administrativa Oficial de Portugal, obtained from the Direcção-Geral de Território) and proceeded to subset it to only contain the values for the municipality of Pedrogão Grande, then using it to rasterize a spatial file. By passing this spatial file over our data cube, which includes data from the entire Portugal landmass, it will obtain the values for the bands in the pixels which fall under the spatial file extents. For purposes of this case study, we first used NBR to plot an interactive map of Pedrogão Grande, where mouse hovering would return in which class of NBR Fire Severity Scale did that specific area. The classes, color coded, represent the five severity levels presented above, while the area was plotted over a topographic map or Portugal, with zoom, hover, click and drag capabilities.

While in this document it is not possible to submit said interactive map, we can show what it looks like visually. We also plotted the number of sub-areas/pixels were placed under which class, to gauge the overall impact of the fire in the Pedrogão Grande municipality.

While this visualization has a meaningful purpose in the evaluation of the spread and impact of the fire, it has the limitation of being currently restricted to the geographical delimitations in the original shapefile being bigger than the spatial extent of each pixel. This means that each subsection of the images, corresponding to the “freguesias” of the municipality of Pedrogão Grande, return values corresponding to the aggregation of 3 to 5 pixels of the MODIS data, depending on the size. While not always a problem, there are cases in which we may need to represent the data as close to the original as possible, which would mean we would have to use and build different shapefiles for the purposes, where the delimitation is made on a 1x1 km grid to align with the satellite data, in the MODIS case.

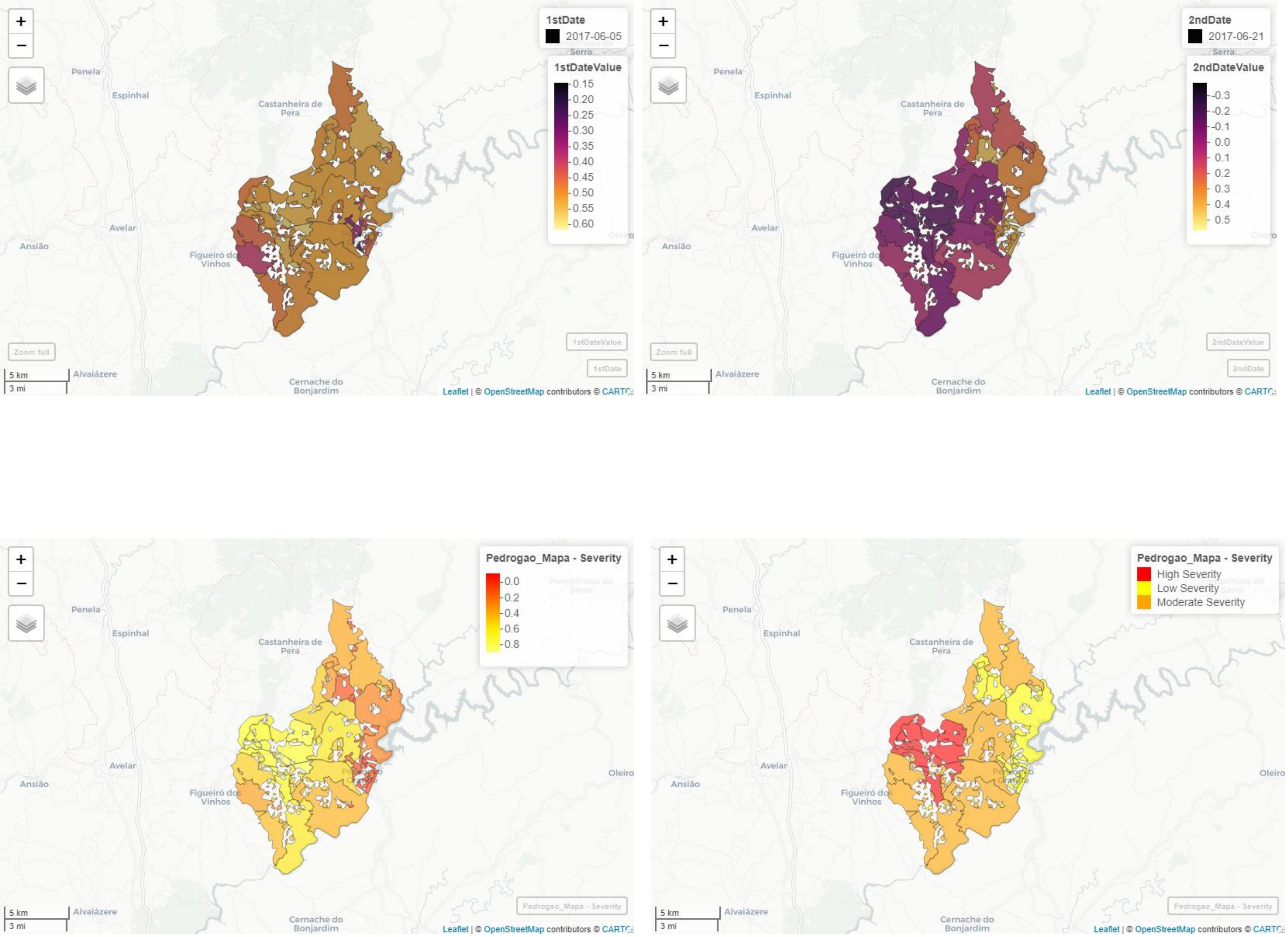


Figure 9 From left to right and top to bottom, Pedrogão municipality map with the NBR values pre-fire, post-fire, the delta values between both instances, and the classification of each area as an impacted area by fire, derived from the DeltaNBR values.

At this point, spectral metadata obtained from satellite imaging has been processed and ran through a series of scripts until we have obtained the fire impact classification of a pre-determined area, according to peer-reviewed spectral indexes computation. From here on forward we now can also take the same processed metadata, and instead of focusing on a more visual and interactive side, we can also use that data to construct time-series of spectral indexes (table 9), both for first-hand monitoring and assessment, but also functioning as base for future machine learning models and time-series evaluation packages when used in pair with in-situ data and other instruments of monitoring.

```
PedrogãoNBR = select_bands(Portugal_Cube_LSR, c("B02","B07")) %>%
   apply_pixel("((B02-B07)/(B02+B07))", names = "NBR") %>%
   extract_geom(z, FUN=mean)

PedrogãoNDVI = select_bands(Portugal_Cube_LSR, c("B01","B02")) %>%
   apply_pixel("(B02-B01)/(B02+B01)", names = "NDVI") %>%
   extract_geom(z, FUN=mean)

PedrogãoSAVI = select_bands(Portugal_Cube_LSR, c("B01","B02")) %>%
   apply_pixel("1.5*((B02-B01)/(B02+B01+0.5))", names = "NDVI") %>%
   extract_geom(z, FUN=mean)
```

Table 9 Creation of time-series derived from data obtained from the Pedrogão Area shapefile. Time delineation was omitted but referenced in the text. Output is a dataframe of values for each spectral index.

And so, now that we established the impact of the fire, using the Normalized Burn Ratio, we want to assess the long-term impact and recovery of the vegetation soil and surface in the area. For that, we first obtain the values for three different time-series for the target area (table 9).

These 3 functions return a data frame with the values for every pixel in the spatial file “z” (Pedrogão), for every occasion where the satellite measured it from t0 to t1, both of which user decided. We settled on June 6th, 2016, all the way to January 1st, 2020, so that we could check initial pre-fire values for all indexes, the impact the fire had on the indexes, and the recovery process for the area. Since in this instance, we are only exploring the data for the area of itself, with no particular interest in any individual pixel, we chose to group the observations by time, using the mean value of all pixels for that individual time. So instead of n observations on x day with y value, we have 1 observation on X Day, with value equal to the mean of the n observations. The time variable was converted to date format, as it is provided simply as a string of characters in its original form. This was executed for the 4 indexes and correspondent data frames (table 10 and 11).

```
NBRPedrogão <- data.frame(PedrogãoNBR)
NBRPedrogão <- NBRPedrogão %>% group_by(time) %>% summarize(NBRValue = mean(NBR,
na.rm = TRUE))
NBRPedrogão <- as.data.frame(NBRPedrogão)
NBRPedrogão$time <- as.Date(NBRPedrogão$time, format="%Y-%m-%d")
```

Table 10 Data frame manipulation for ensuing visualization of the time-series

```
PlotSingleTimeSeries_SAVI <- ggplot(data=SAVIPedrogão,
            aes(x=time, y=SAVI)) +
  geom_line( linetype = 1, size = 1) +
  ylim(0,2)
print(PlotSingleTimeSeries_SAVI + ggtitle("SAVI time-series in the Pedrogão Area"))


#Plotting 3 timeseries

df_list <- list(NDVIPedrogão, NBRPedrogão, SAVIPedrogão)
df_list %>% reduce(full_join, by='time')
TimeSeriesLongFormat <- melt(df_list, id="time")  # convert to long format

p <- ggplot(data=TimeSeriesDFLong,
       aes(x=time, y=value, colour=variable)) +
  geom_line( linetype = 1, size = 1) +
  ylim(0,2)
print(p + ggtitle("Time-Series of 3 spectral indices in the Pedrogão Area"))
```

Table 11 Merging of each singular data-frame and posterior visualization arguments

At this point we have 3 different data frames, composed of a time-series of its respective spectral index. Two pathways are available, either plotting each of them singularly for easier visualization and data exploration of that index or plotting the three timeseries all together. We will show the code for either version but only plotted the 3 time-series together, for the sake of comparison and visualization.

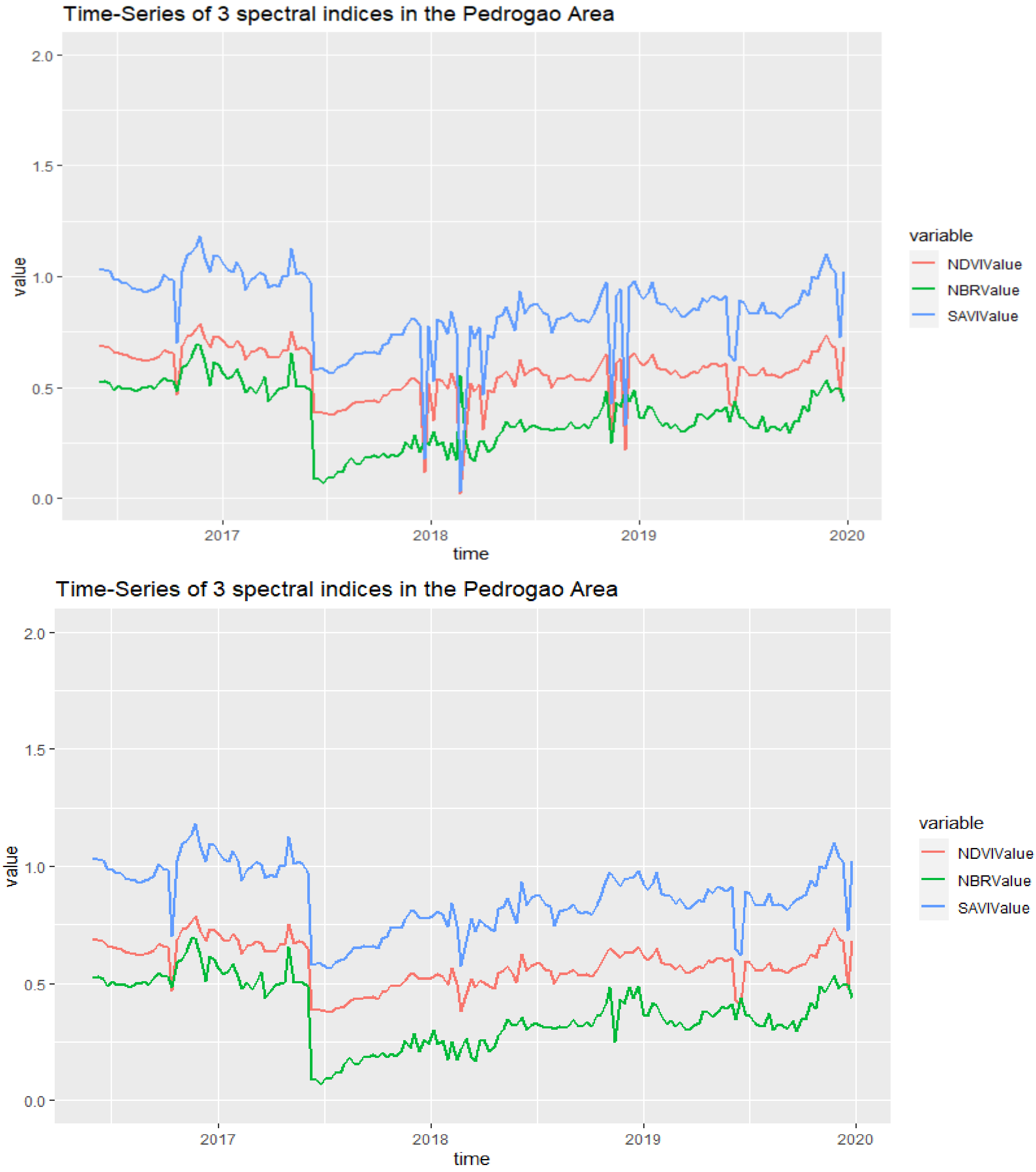


Figure 10 non-Smoothed (top) and Smoothed (bottom) Time-Series of the Pedrogão Área for three spectral indexes

As we can see in the figure, after the instant severe decline in the vegetation indexes and NBR value as the fires happened, almost 3 years after the fact there was still only an approximation to the pre-fire values with none of them reaching them on a consistent basis, showing the lasting damage that the occurrence had in the area. This is consistent with the literature, in which 4 to 8 years is the consistent as a timeframe for the vegetation "greenness" indexes to return to pre-fire levels. We decided to add a smoothed version of the time-series, as there were present negative and positive peaks due to mis-observations of the band values in certain dates. We did it by replacing those mis-observations by the average values of the previous and posterior observation.

We also decided to add a secondary query and timeseries analysis process, where we plot the time-series for the NDVI values across the same time span but split by severity levels.

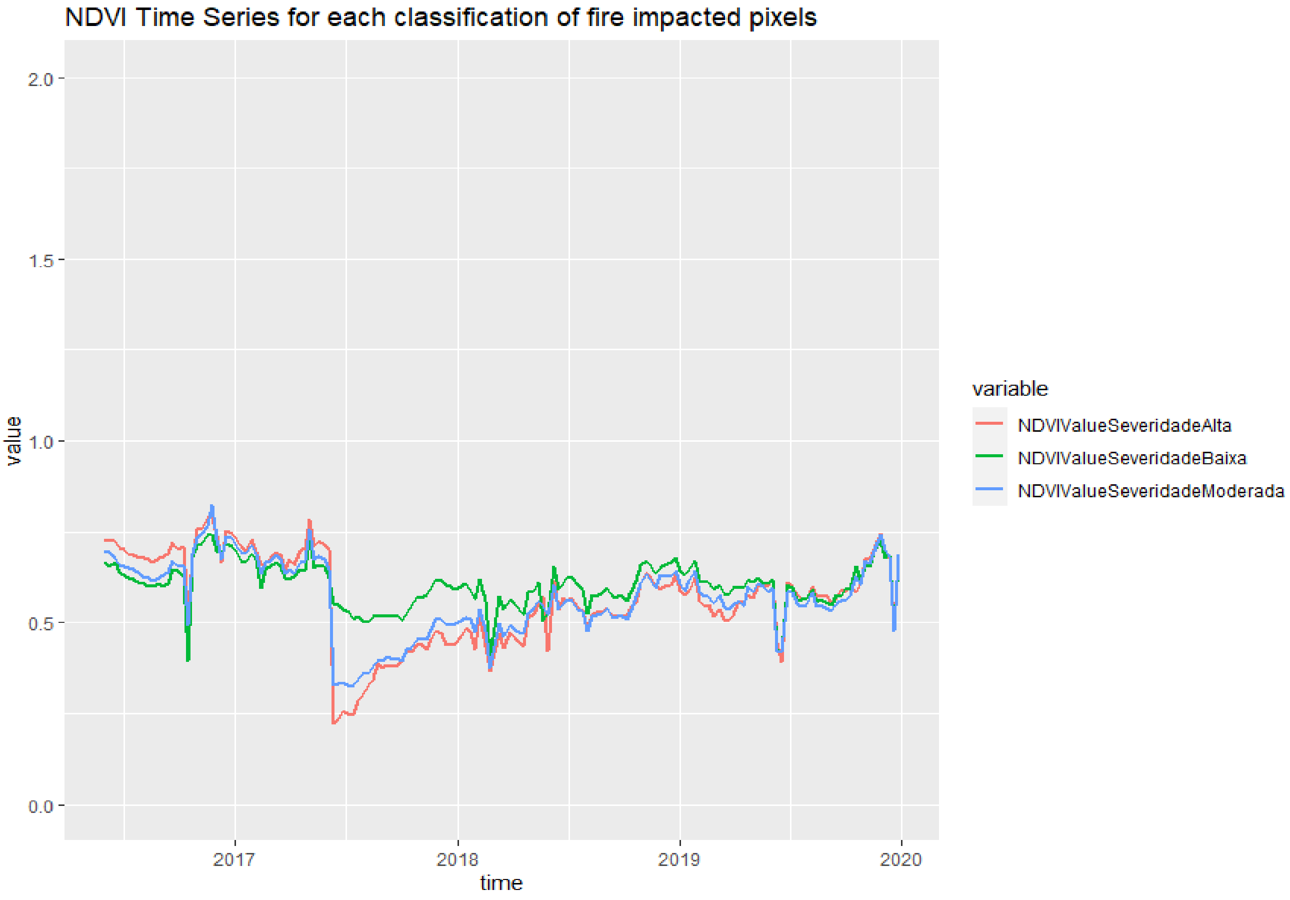


Figure 11 NDVI values split by areas with distinct severity impact levels, as calculated before.

As we can see, areas which were classified as high severity of impact had a more abrupt decline of NDVI values, leading to a higher percentual increase of NDVI in the following years, taking a bigger effort into achieving pre-fire levels or values close to it. Low severity areas have a noticeably less relevant impact at the point of fire, and as such have a smoother recuperation line on the time series, as there's less of a slope needed to recover regarding the index values.

3.4. Performance Evaluation

Considering land surface products as well as multiple water (Sea Surface Temperature, Ocean Reflectance, Particulate Organic Carbon Concentration), and atmospheric resources (Aerosol Product, Precipitable Water) also available for future ingestion and processing, there are thirty-seven different available MODIS products. In addition to these, the pipeline is also designed to ingest major Sentinel-2 products. While the ingestion of all of them is not the endgame of the project, as many of them are different day composites of the same product (daily composites vs 8-day composites), what is currently seventy gigabytes of data representing three products for the span of ten years can easily be expected to become terabytes of data in a near future. As such, understanding the strain that an increase in size represents to the data cube/pipeline is of the utmost importance.

For this purpose, we designed multiple time-evaluating performance tests. After exploration, we understood that the biggest computing weight of the process if the creation of the data cube itself, especially the image collection creation, using the ingestion .json files and the post-download MODIS products. For these tests, we ran two different datasets, one with two years' worth of data from the three different MODIS products, as this is the minimum dataset required to be able to make any year-to-year claim of environmental or biodiversity claims (further referenced and explored) and a second dataset with ten years' worth of data, to act as a variable when measuring the scalability of our pipeline.

The MODIS download part of the pipeline is inherently harder to measure as it is dependent on the network behind the download, and such internet is more complex and variable than CPU and script performance measurement. Even so, the download of 70.1 gigabytes of data referring to the 10-year span used in the stress and scalability testing took a total of 5 hours and 38 minutes, made using a powerline to directly connect to a NOS router set for 240MB/s default speed. As previously stated, this time is dependent on network speed at a local level (inherently variable by sheer virtue of peak hours for traffic, amongst others influences, as well as interferences in the HTTPS server used for download. All downloads were made from the same server, https://e4ftl01.cr.usgs.gov/ .

After measuring the performance of the data cube creation phase of the pipeline, we also designed multiple environmental and biodiversity relevant queries to the merged data cube with the three products, simultaneously measuring the capacity of the pipeline to obtain specific queries, as well as showing the diversity and various possibilities of queries available from the built data cube. We then ran these queries multiple times, to record the time taken by each query/request.

.

3.5. Data Cube Building Computing Performance

Using a simple repetition script recording the time each iteration takes, we made it so three different expressions were benchmarked. The three correspond to the Data Cube Creation for the different MODIS products, over the same spatial extent, Portugal Continental territory. Split into two phases, the first measured the creation of the cubes for a two-year time span, the 2nd one measure the creation of the cubes for a ten-year span. They were evaluated 100 times each, and while the 3 of them were measured at once for logistic purposes, the benchmarking was made with random order within each other, to avoid any possible tendencies or hidden learning. To show the results, we plotted the distribution of the one hundred evaluations, in seconds, as well as a statistic summary of the benchmarking. As expected, we noticed an increase in the ingestion time, by a factor of six for each separated cube. The merging of the posterior cubes is a under 0.1 second function, given it functions as a simple data structure merge by a spatio-temporal extent.

With this into account, the creation of the three different MODIS products data cubes, with the bands from the three original products under the same spatial extent, takes 132 seconds (2 minutes and 12 seconds) when given a two-year span of available data vs 669 seconds (11 minutes and 9 seconds) when given a ten-year span. While this shows the time constraint of dealing with an increased amount of data, it is to be remembered that when talking about the ability of a user to create a data cube, this is a one-time action for a use of an overly arching data cube.

As far as the creation of a National Data Cube, the idea behind the pseudo-API revolves around after the first creation and hosting of the Data Cube, subsequent merges would only encapsulate newly obtained MODIS files, at a weekly or monthly rate, diminishing the required processing time required with each update. The first creation would be the most demanding one, which we estimate would require up to 40 minutes, assuming a first build with the currently in use products and the earliest available data, from the year 2000, all the way to the most recently available data.

Cube	Min(s)	Lower quartile(s)	Mean(s)	Median(s)	Upper quartile(s)	Max(s)	Number of test runs
MODIS09A1TwoYears	71.43	72.19	72.68	72.59	73.01	75.67	100
MODIS11A2TwoYears	51.01	52.21	52.57	51.54	52.86	55.40	100
MODIS14A2TwoYears	6.11	6.22	6.29	6.29	6.35	6.59	100
MODIS09A1TenYears	354.91	258.19	370.48	360.23	374.07	439.13	100
MODIS11A2TenYears	251.19	260.85	268.05	262.41	269.55	324.31	100
MODIS14A2TenYears	30.35	30.67	31.63	30.827	31.51	41.17	100

Table 12 Performance statistics for the ingestion of metadata and creation of data-cubes for each MODIS product

As we can see by the benchmarking results in table 12, different MODIS products produce vastly differing results. This is explained by the different in original file size to be ingested. MODIS09A1 downloaded files amounted to 64.5 GB of data, while the same temporal extent for MODIS14A2 files produced 570 MB of data. Such effect stems from the differing number of bands/layers included in each file, as well as the resolution at which the original instrument obtained the data. MODIS09A1 data is collected at a 500 m base spatial resolution while MODIS11A2 and MODIS14A2 are collected at 1km base spatial resolution. These two are the main reasons behind the distinct data size. The better the resolution and larger the number of bands, the larger will the downloaded data folder be, and consequently the following ingestion process into the data cube. The violin plots designed used a log scale for easier visualization.

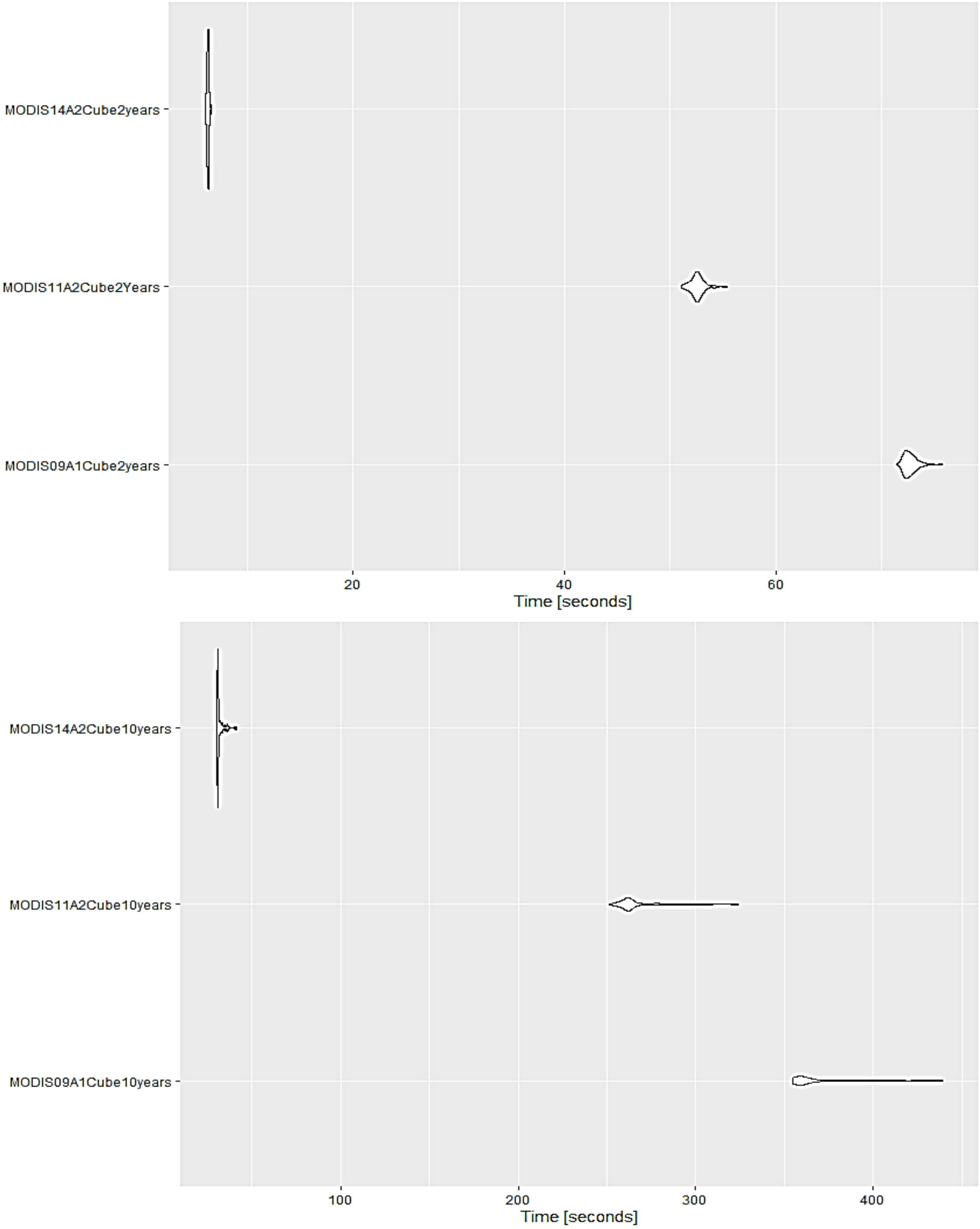


Figure 12 Violin Plots for the previously established performance statistics. Plot used a logarithmic scale for easier visualization

3.6. Environmental Queries Computing Performance

In the next performance testing phase, we designed a series of tests to measure the performance behind the environmental queries referred in the previous section, for the LST and NDVI, as well as similar queries for EVI values, another spectral index. This was made to ensure the number of bands required for the calculus of a spectral index didn't severely impact the computing time of that spectral index.

First up, we computed the performance testing for a series of plots visualizing the spatial distribution for values of three assorted primary and secondary products, the LST (**Land Surface Temperature**), the NVDI (**normalized difference vegetation index**) and the EVI (**Enhanced Vegetation Index)**.

Divided in two stages, we first measured their computing times with a spatial resolution of 1 km per pixel, the highest possible resolution allowed with our product combination.

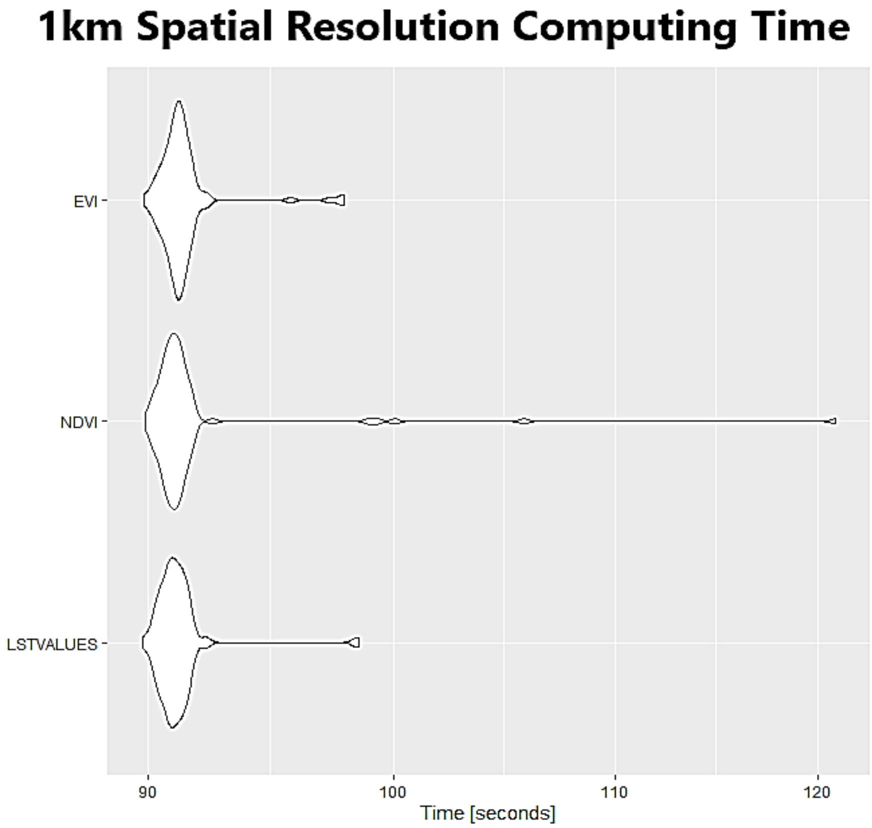


Figure 13 Violin Plots for the computation time of 1k spatial resolution group of plots.

.

On simplified terms, from the moment we pass the query to the data cube structure, with the code referenced in the results section referent to the plot creation, it takes an average of slightly over 90 seconds for it to obtain six 1 km spatial resolution of a determined band on six separate occasions amongst the datasets. This is true for every band, every date selection. We noticed the increment in the number of images request led to a linear increment in the computing time, with each singular image added to the query representing between 14 to 16 seconds of added computing time. As soon as data cube access is granted, it takes around 15 seconds to obtain any image from any date as well as the possibility of saving it under .png or .jpg formats.

At the next stage of the performance testing, we then decided to see how resolution impacted the computing time of the plots. As such, we created a “copycat” cube, with the spatial resolution set to 5 km per pixel. We noticed an immense gain in script performance, which would indicate that for queries that do not require a high degree of spatial resolution,

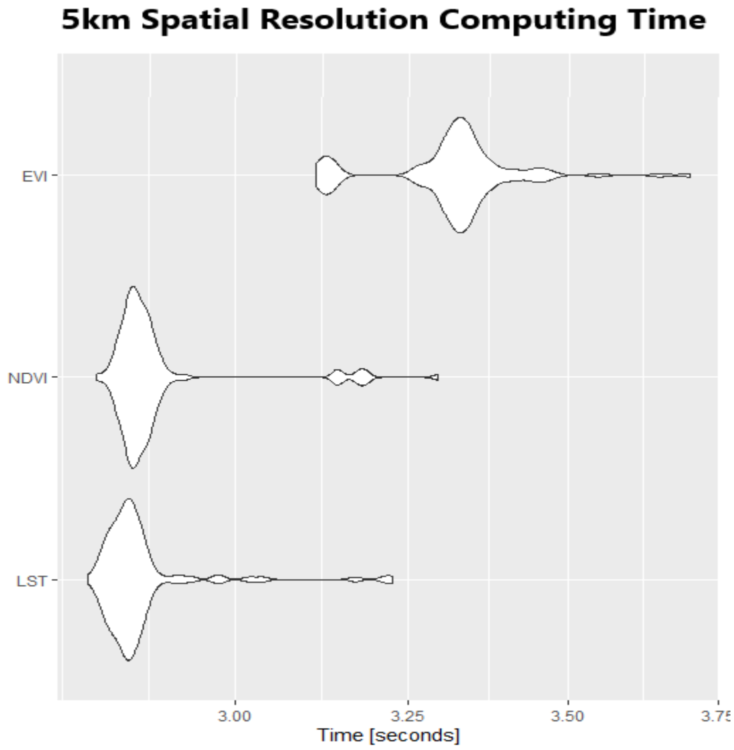


Figure 14 Computing performance for a 5 km spatial resolution cube

5 km or similar spatial resolutions would be a great performance enhancing tool for the running scripts.

Finally, and as an empiric way to prove what we thought was the likely scenario in the code format being used, we decided to test whether the computing order affected the total time used for the script to run. As such, we decided to test 2 versions of the same EVI query, one where we first subset the data cube by the pretended bands and only after that do we scan through it for the desired dates, and another one where we reversed the order.

As expected, the cube where the bands were selected first had a better computing performance. This is due to the data cube layered structure, by automatically subsetting it to only require values for the desired numbers of bands N, in which N < the total number of bands present in the data cube, the posterior date search will run over a lesser number of "layers" of the cube.

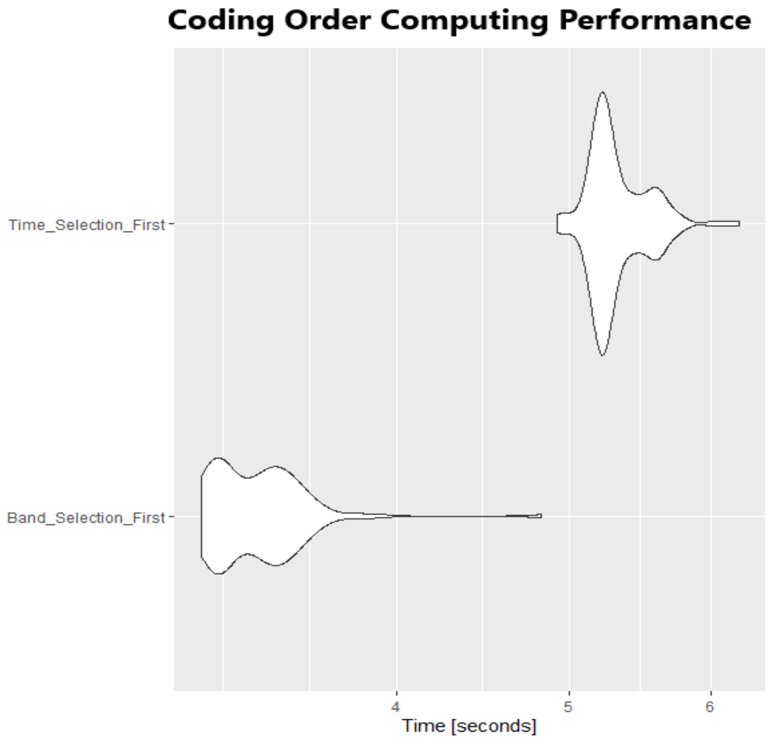


Figure 15 Difference in computation times depending on the coding order

3.7. Case Study Computing Performance

Finally, and as the main example of the possibilities behind the data cube structure as currently designed, we computed the performance of the entire pipeline behind the case study previously presented as a single block, which means it represents the time taken from the extraction of the desired analysis ready data for the desired location and timeframe. We did not count the ingestion/building of the data cube as part of the pipeline as the assumption is made that end users would not have to go through that part of the process, and we already performed the computing performance tests for that part of the pipeline. While the example taken here is towards Pedrogão Grande, comparable results are obtained for any municipality or district, with a small variation depending on the total study area. The pipeline as described here represents the processing needed to obtain both the interactive map with the Delta NBR values and the histograms assessing the severity of the fire impact, as well as the time-series produced, both singular and the plotting of the 4 time-series at the same time. As with previous performance tests, continuous attempts of improving the code efficiency by refactoring are being made.

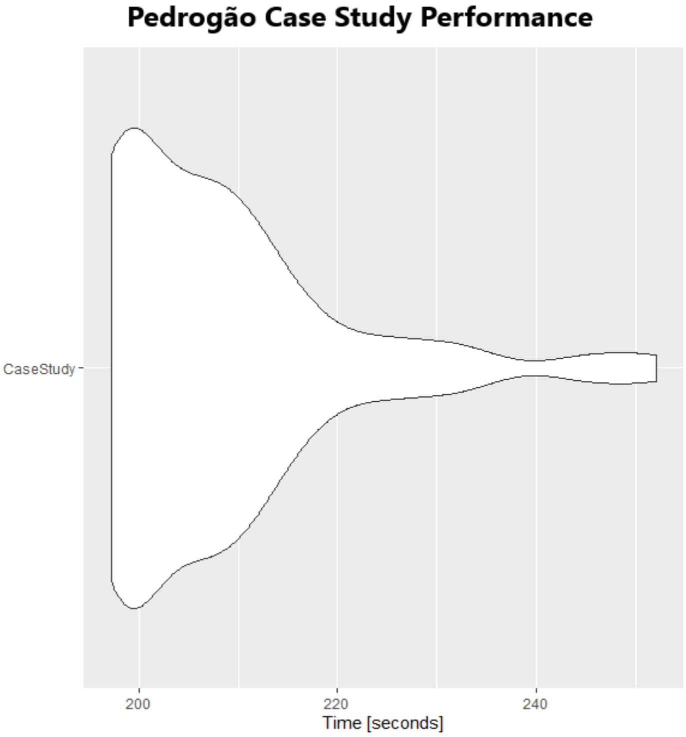
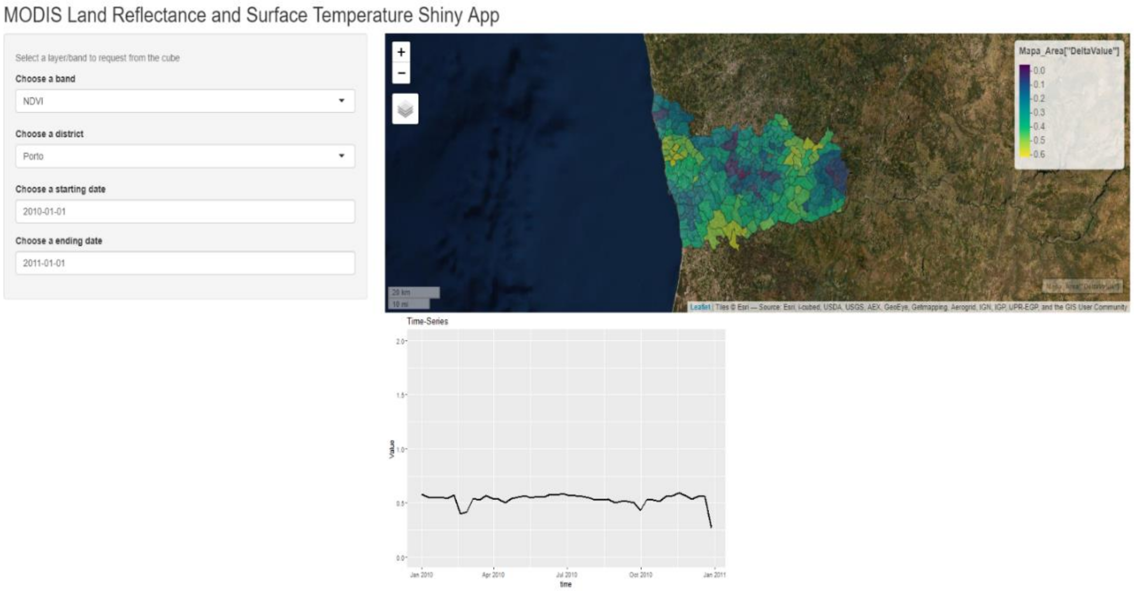


Figure 16 Computing time for the entire case study code pipeline

3.8. Visual Interface

One of the main and final objectives we have with the project is the creation of an end-user applicable visual interface that allows for the user to submit personal and specific requests to the data cube involving structure, returning the pretended output. We did this using the shiny app package and side packages to help with the visualization and at this stage, there is a robust and deployable visual interface ready to go public. Nonetheless, continuous efforts are being made to add and improve features to the application, for a broader reach and versatility of outputs, so that the maximum amount of end uses may be tackled. Given the document format, we took some screenshots to allow for a better understanding of the composition of the interface and possibilities attached to it.



MODIS Land Reflectance and Surface Temperature Shiny App

Select a layer/band to request from the cube

Choose a band: NDVI

Choose a district: Porto

Choose a starting date: 2010-01-01

Choose a ending date: 2011-01-01

+
−
Mapa_Area["DeltaValue"]
0.0
0.1
0.2
0.3
0.4
0.5
0.6
20 km
10 mi
Leaflet | Tiles © Esri — Source: Esri, i-cubed, USDA, USGS, AEX, GeoEye, Getmapping, Aerogrid, IGN, IGP, UPR-EGP, and the GIS User Community
Time-Series
2.0
1.5
1.0
0.5
0.0
Value
Jan 2010
Apr 2010
Jul 2010
Oct 2010
Jan 2011
time

Figure 17 Visual interface prototype

In the picture above, we see a query for the NDVI index values across the Porto district, between January 1st 2010 and January 1st 2011. This produces two automatic outputs, a map representing the value difference between those days, which may be used to assess the impact of punctual unexpected events in a certain area, using index values such as NBR for fire impact, Flood Index, Snow indexes to understand extent, amongst others. The second output is a time series of values for the requested index/band across the entire timespan in the input, in the delimitated area, which may be used to produce time-series for specific areas when affected by long-term events like droughts, climate change, amongst others. While in this demonstration interface, we used a reduced timespan, for computation time purposes, the code behind the application was tested for scalability, and as such can be used in the ulterior data cube with the complete timespan as ingested.

4. Conclusion, Limitations and Future Work

The purpose of this project and dissertation was to lay the groundwork for the definite construction of a Portuguese National Data cube, by showing and demonstrating a working pipeline allowing the download, processing, and ingestion of data into a data structure whose main selling point is the adaptability and capability of holding multispectral data in one base structure. As we close this part of the project, there is a basis of work established from which further steps can be taken, by taken advantage of the scripts prepared and refactored, scripts whose interoperability with other programming languages other than R provide scalability and adaptability to the pipeline as the market and area of Data Cubes as a mainstream data structure grows. While in this project we used a 10-year sample data cube as display of the proof-of-concept and for all the programming performance testing done in this dissertation, as currently structured, the code is ready for deployment of a 22-year data-cube, from 2000 to 2022 for available products. As referred in the respective section, for products not supported by the base package *gdalcubes*, we constructed an ingestion file for additional products used (MOD09A1). If there was to be an immediate (pre-2023) release of a Portuguese Data Cube back-end structure, ingestion files are supported for 14 different open-source free satellite imaging products, 10 MODIS products and 4 Sentinel-2 products. Ingestion of other MODIS and Sentinel products is also not impossible or particularly difficult, as both projects publicly share the structure of their metadata files, allowing us to interpolate the structure that the ingestion file, always in .json must take for the pipeline to work.

The project is also not meant to be purely a back-end data structure process. The intent is to support said backend with a front-end visual interface, allowing for users to make a set of selected inputs, such as time, multispectral band, arithmetic functions to be applied, as well as spatial polygon drawing on maps itself, allowing for spatial subsetting with mouse clicks or coordinate inputs. While R isn’t the preferred choice of a front-end hub, it is still a useful one, which we have explored by making smaller shiny apps running and publicly available, showing the purposefulness and usefulness of the back-end structure to support environmental queries.

As an overall conclusion on this stage of process and project, there is definitely enough support, both from a biological and a programming perspective for building upon this project with an intent to grow it from an available data standpoint, creating a central hub to act as the host for the MODIS metadata and the running and refactored R code to run the pipeline at a scheduled routine, maintaining a continuous 24 hours per day up and running front end visual interface. This would allow for it to work as a hub for environmental and biodiversity monitoring data queries and possible start point for academic paper data and visualizations.

As far as performance goes, the times obtained for environmental queries show tolerable and even desirable computing times for mainstream queries, and while at an early stage, there is a clear scalability and variability value in the project and code that allows for quick shifts in queries by location, time, bands, that can’t currently be quantified by any existent coding or done by existent workflows. That remains the big performance advantage of the data cube structure when put against more traditional methods for time-series creations and specific geo-spatial queries.

There are current limitations that stop this structure from being a be all and end all product that would instantly take-over the area. Perhaps the most overarching and the more “code” focused one is that while it is well on its way to become **the** language of choice for the treatment of spatial data, R has a few inherent limitation and problems. On the front-end, R, and shiny applications, aren’t tailored as websites and web endpoints, having a big, but still limited pool of available widgets, html and css functions when compared to the popular front-end languages. Secondly, and perhaps a bridge into a more logistical discussion to be taken into the future, R only allows for *in-memory* work, which means you’ll always be limited by the RAM of the current workstation. This poses a problem depending on the final purpose of the project. If it were ever to transition to a commercial approach, then adaptation towards cloud computing would be required as a workaround for the problem.

As an academical learning application, the project has a lot of potential for mainstream use, as it is becoming more and more apparent with the ever-growing number of large-scale data cubes being built and upkept. There are frameworks public for the launching of the final version of such a product, and the Swiss model could perhaps show to be a good initial point, by making the Data Cube contents, both at back-end and front-end level, available to learning institutions and possibly strict organizations tasked with monitoring and biodiversity modelling. It also can work as a framing for the temporal timeline of the rest of the project. The SDC has been in early planning and testing phase since 2016, and is currently still building on those first tests, running a prototype application. This is now the next challenge for the Portuguese Data Cube. We established an early proof-of-concept of the advantages and disadvantages of the cube as well as delineated an overarching working pipeline for its implementation on a larger scale, and we should now begin conversations over the logistics and structure behind the code. Organizations and licenses, wide-spread vs localized first testing phases and extent of the first public prototype. These are all decisions which now need to be argued and discussed in order to define the immediate next steps.

References

[1] A. Losch, “The need of an ethics of planetary sustainability,” *Int J Astrobiol*, vol. 18, no. 3, pp. 259–266, 2019, doi: 10.1017/S1473550417000490.

[2] N. K. Arora, “Environmental Sustainability—necessary for survival,” *Environmental Sustainability*, vol. 1, no. 1, pp. 1–2, 2018, doi: 10.1007/s42398-018-0013-3.

[3] H. Guo, Z. Liu, H. Jiang, C. Wang, J. Liu, and D. Liang, “Big Earth Data: a new challenge and opportunity for Digital Earth’s development,” *Int J Digit Earth*, vol. 10, no. 1, pp. 1–12, 2017, doi: 10.1080/17538947.2016.1264490.

[4] D. Xu, Y. Ma, J. Yan, P. Liu, and L. Chen, “Spatial-feature data cube for spatiotemporal remote sensing data processing and analysis,” *Computing*, vol. 102, no. 6, pp. 1447–1461, 2020, doi: 10.1007/s00607-018-0681-y.

[5] D. Boyd and K. Crawford, “Critical questions for big data: Provocations for a cultural, technological, and scholarly phenomenon,” *Inf Commun Soc*, vol. 15, no. 5, pp. 662–679, 2012, doi: 10.1080/1369118X.2012.678878.

[6] A. Lewis *et al.*, “The Australian Geoscience Data Cube — Foundations and lessons learned,” *Remote Sens Environ*, vol. 202, pp. 276–292, 2017, doi: 10.1016/j.rse.2017.03.015.

[7] S. Rivest, Y. Bédard, M. J. Proulx, M. Nadeau, F. Hubert, and J. Pastor, “SOLAP technology: Merging business intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 60, no. 1, pp. 17–33, 2005, doi: 10.1016/j.isprsjprs.2005.10.002.

[8] T. Esch *et al.*, “Exploiting big earth data from space – first experiences with the timescan processing chain,” *Big Earth Data*, vol. 2, no. 1, pp. 36–55, 2018, doi: 10.1080/20964471.2018.1433790.

[9] M. Sudmanns *et al.*, “Big Earth data: disruptive changes in Earth observation data management and analysis?,” *International Journal of Digital Earth*, vol. 13, no. 7. Taylor and Francis Ltd., pp. 832–850, Jul. 02, 2020. doi: 10.1080/17538947.2019.1585976.

[10] S. Nativi, P. Mazzetti, and M. Craglia, “A view-based model of data-cube to support big earth data systems interoperability,” *Big Earth Data*, vol. 1, no. 1–2, pp. 75–99, 2017, doi: 10.1080/20964471.2017.1404232.

[11] M. Appel and E. Pebesma, “On-demand processing of data cubes from satellite image collections with the gdalcubes library,” *Data (Basel)*, vol. 4, no. 3, Sep. 2019, doi: 10.3390/data4030092.

[12] P. Baumann *et al.*, “Big Data Analytics for Earth Sciences: the EarthServer approach,” *Int J Digit Earth*, vol. 9, no. 1, pp. 3–29, 2016, doi: 10.1080/17538947.2014.1003106.

[13] M. C. A. Picoli *et al.*, “Big earth observation time series analysis for monitoring Brazilian agriculture,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 145, no. August, pp. 328–339, 2018, doi: 10.1016/j.isprsjprs.2018.08.007.

[14] G. Giuliani *et al.*, “Building an Earth Observations Data Cube: lessons learned from the Swiss Data Cube (SDC) on generating Analysis Ready Data (ARD),” *Big Earth Data*, vol. 1, no. 1–2, pp. 100–117, 2017, doi: 10.1080/20964471.2017.1398903.

[15] D. Alcaraz-Segura *et al.*, “Potential of satellite-derived ecosystem functional attributes to anticipate species range shifts,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 57, pp. 86–92, 2017, doi: 10.1016/j.jag.2016.12.009.

[16] C. C. Lautenbacher, “The Global Earth Observation System of Systems: Science serving society,” *Space Policy*, vol. 22, no. 1, pp. 8–11, 2006, doi: 10.1016/j.spacepol.2005.12.004.

[17] CEOS, “EO Handbook.” http://database.eohandbook.com/database/missiontable.aspx (accessed Aug. 09, 2022).

[18] J. T. Overpeck, G. A. Meehl, S. Bony, and D. R. Easterling, “Climate data challenges in the 21st century,” *Science*, vol. 331, no. 6018. pp. 700–702, Feb. 11, 2011. doi: 10.1126/science.1197869.

[19] T. João, G. João, M. Bruno, and H. João, “Indicator-based assessment of post-fire recovery dynamics using satellite NDVI time-series,” *Ecol Indic*, vol. 89, no. January, pp. 199–212, 2018, doi: 10.1016/j.ecolind.2018.02.008.

[20] E. F. Vermote, J. C. Roger, and J. P. Ray, “MODIS Surface Reflectance User’s Guide Correspondence e-mail address: mod09@ltdri.org,” 2015. [Online]. Available: http://modis-sr.ltdri.org

[21] EPSG Geodetic Parameter Dataset, “EPSG Portugal Code.” https://epsg.io/3763 (accessed Apr. 08, 2022).

[22] E. J. Weyuker, “Experience with performance testing of software systems: issues, an approach, and case study,” *IEEE Transactions on Software Engineering*, vol. 26, no. 12, pp. 1147–1156, Dec. 2000, doi: 10.1109/32.888628.

[23] Z. L. Li *et al.*, “Satellite-derived land surface temperature: Current status and perspectives,” *Remote Sensing of Environment*, vol. 131. pp. 14–37, Apr. 05, 2013. doi: 10.1016/j.rse.2012.12.008.

[24] J. R. Townshend *et al.*, “The 1 km resolution global data set: Needs of the international geosphere biosphere programme!,” *Int J Remote Sens*, vol. 15, no. 17, pp. 3417–3441, 1994, doi: 10.1080/01431169408954338.

[25] Y. Li *et al.*, “Potential and actual impacts of deforestation and afforestation on land surface temperature,” *J Geophys Res*, vol. 121, no. 24, pp. 14372–14386, Dec. 2016, doi: 10.1002/2016JD024969.

[26] D. J. Mildrexler, M. Zhao, W. B. Cohen, S. W. Running, X. P. Song, and M. O. Jones, “Thermal anomalies detect critical global land surface changes,” *J Appl Meteorol Climatol*, vol. 57, no. 2, pp. 391–411, Feb. 2018, doi: 10.1175/JAMC-D-17-0093.1.

[27] S. Huang, L. Tang, J. P. Hupy, Y. Wang, and G. Shao, “A commentary review on the use of normalized difference vegetation index (NDVI) in the era of popular remote sensing,” *Journal of Forestry Research*, vol. 32, no. 1. Northeast Forestry University, Feb. 01, 2021. doi: 10.1007/s11676-020-01155-1.

[28] Y. Ke, J. Im, J. Lee, H. Gong, and Y. Ryu, “Characteristics of Landsat 8 OLI-derived NDVI by comparison with multiple satellite sensors and in-situ observations,” *Remote Sens Environ*, vol. 164, pp. 298–313, Jul. 2015, doi: 10.1016/j.rse.2015.04.004.

[29] L. M. Ribeiro, A. Rodrigues, D. Lucas, and D. X. Viegas, “The impact on structures of the pedrógão grande fire complex in June 2017 (Portugal),” *Fire*, vol. 3, no. 4, pp. 1–22, Dec. 2020, doi: 10.3390/fire3040057.

[30] Z. Jiang, A. R. Huete, K. Didan, and T. Miura, “Development of a two-band enhanced vegetation index without a blue band,” *Remote Sens Environ*, vol. 112, no. 10, pp. 3833–3845, Oct. 2008, doi: 10.1016/j.rse.2008.06.006.

[31] N. Abdalla, N. I. Abdalla, A. Karamalla Gaiballa, C. Kätsch, M. Sulieman, and A. Mariod, “Using MODIS-Derived NDVI and SAVI to Distinguish Between Different Rangeland Sites According to Soil Types in Semi-Arid Areas of Sudan (North Kordofan State),” 2015. [Online]. Available: http://www.aiscience.org/journal/ijlsehttp://creativecommons.org/licenses/by-nc/4.0/

[32] A. Teodoro and A. Amaral, “A statistical and spatial analysis of portuguese forest fires in summer 2016 considering landsat 8 and sentinel 2A data,” *Environments - MDPI*, vol. 6, no. 3, Mar. 2019, doi: 10.3390/environments6030036.

Annexes

```
#!/bin/bash
GREP_OPTIONS=''

cookiejar=$(mktemp cookies.XXXXXXXXXX)
netrc=$(mktemp netrc.XXXXXXXXXX)
chmod 0600 "$cookiejar" "$netrc"
function finish {
  rm -rf "$cookiejar" "$netrc"
}

trap finish EXIT
WGETRC="$wgetrc"

prompt_credentials() {
    echo "Enter your Earthdata Login or other provider supplied credentials"
    read -p "Username (nuno_sousa): " username
    username=${username:-nuno_sousa}
    read -s -p "Password: " password
    echo "machine urs.earthdata.nasa.gov login $username password $password" >> $netrc
    echo
}

exit_with_error() {
    echo
    echo "Unable to Retrieve Data"
    echo
    echo $1
    echo
    echo
"https://e4ftl01.cr.usgs.gov//DP131/MOLT/MOD09A1.061/2017.08.05/MOD09A1.A2017217.h17v05
.061.2021280020745.hdf"
    echo
    exit 1
}

prompt_credentials
  detect_app_approval() {
    approved=`curl -s -b "$cookiejar" -c "$cookiejar" -L --max-redirs 5 --netrc-file
"$netrc"
https://e4ftl01.cr.usgs.gov//DP131/MOLT/MOD09A1.061/2017.08.05/MOD09A1.A2017217.h17v05.
061.2021280020745.hdf -w %{http_code} | tail  -1`
    if [ "$approved" -ne "302" ]; then
        # User didn't approve the app. Direct users to approve the app in URS
        exit_with_error "Please ensure that you have authorized the remote application
by visiting the link below "
    fi
}

setup_auth_curl() {
    # Firstly, check if it require URS authentication
    status=$(curl -s -z "$(date)" -w %{http_code}
https://e4ftl01.cr.usgs.gov//DP131/MOLT/MOD09A1.061/2017.08.05/MOD09A1.A2017217.h17v05.
061.2021280020745.hdf | tail -1)
    if [[ "$status" -ne "200" && "$status" -ne "304" ]]; then
        # URS authentication is required. Now further check if the application/remote
service is approved.
        detect_app_approval
    fi
}

setup_auth_wget() {
    # The safest way to auth via curl is netrc. Note: there's no checking or feedback
    # if login is unsuccessful
    touch ~/.netrc
    chmod 0600 ~/.netrc
    credentials=$(grep 'machine urs.earthdata.nasa.gov' ~/.netrc)
    if [ -z "$credentials" ]; then
        cat "$netrc" >> ~/.netrc
    fi
}

fetch_urls() {
  if command -v curl >/dev/null 2>&1; then
      setup_auth_curl
      while read -r line; do
        # Get everything after the last '/'
        filename="${line##*/}"

        # Strip everything after '?'
        stripped_query_params="${filename%%\?*}"
        curl -f -b "$cookiejar" -c "$cookiejar" -L --netrc-file "$netrc" -g -o $stripped_query_params -- $line && echo || exit_with_error "Command failed with error. Please retrieve the data manually."
      done;
  elif command -v wget >/dev/null 2>&1; then
      # We can't use wget to poke provider server to get info whether or not URS was integrated without download at least one of the files.
      echo
      echo "WARNING: Can't find curl, use wget instead."
      echo "WARNING: Script may not correctly identify Earthdata Login integrations."
      echo
      setup_auth_wget
      while read -r line; do
        # Get everything after the last '/'
        filename="${line##*/}"

        # Strip everything after '?'
        stripped_query_params="${filename%%\?*}"

        wget --load-cookies "$cookiejar" --save-cookies "$cookiejar" --output-document $stripped_query_params --keep-session-cookies -- $line && echo || exit_with_error "Command failed with error. Please retrieve the data manually."
      done;
  else
      exit_with_error "Error: Could not find a command-line downloader.  Please install curl or wget"
  fi
}

fetch_urls <<'EDSCEOF'
https://e4ftl01.cr.usgs.gov//DP131/MOLT/MOD09A1.061/2017.08.05/MOD09A1.A2017217.h17v05.061.2021280020745.hdf
EDSCEOF
```

Table Attachment 1 – Alternative shell script to download MODIS products in case the main framework or package doesn’t work. Obtained from the LP DAAC Data Pool.

```
{
  "selcat": "Radiation Budget Variables - Land Surface Reflectance",
  "selprod": "Surf_Ref_8Days_500m (M*D09A1)",
  "prod_version": "006",
  "sensor": "Terra",
  "bandsel": ["b1_Red", "b2_NIR", "b3_Blue", "b4_Green", "b5_SWIR", "b6_SWIR",
"b7_SWIR", "sur_refl_qc"],
  "quality_bandsel": null,
  "indexes_bandsel": null,
  "download_server": "http",
  "user": "RSCourseCIBIO",
  "password": "Remotesensing123!",
  "downloader": "http",
  "download_range": "Full",
  "start_date": "2010.01.01",
  "end_date": "2020.01.01",
  "spatmeth": "bbox",
  "start_x": 18,
  "end_x": 19,
  "start_y": 0,
  "end_y": 2,
  "bbox": [-121000, -301000, 164000, 278000],
  "spafile": null,
  "drawnext": null,
  "out_projsel": "User Defined",
  "output_proj": "3763",
  "out_res_sel": "Resampled",
  "out_res": 500,
  "resampling": "near",
  "reprocess": false,
  "delete_hdf": false,
  "nodata_change": false,
  "scale_val": false,
  "out_format": "GTiff",
  "compress": "None",
  "out_folder": "C:/Users/nunoe/OneDrive/Ambiente de Trabalho/MODIS09A1",
  "out_folder_mod": "C:/Users/nunoe/OneDrive/Ambiente de Trabalho/MODIS09A1",
  "MODIStspVersion": "2.0.8"
}
```

Table attachment 2 Download file passing arguments for the download of MODIS09A product

```
{
  "description": "Collection format for selected bands from the MODIS MxD09A1
(Aqua and Terra) product",
  "tags": ["MODIS", "surface reflectance"],
  "pattern": ".*\\.hdf.*",
  "subdatasets": true,
  "images": {
    "pattern": "HDF4_EOS:EOS_GRID:\"(.+)\\.hdf.*"
  },
  "datetime": {
    "pattern": ".*M[OY]D09A1\\.A(.{7})[^/]*",
    "format": "%Y%j"
  },
  "bands": {
    "sur_refl_b01": {
      "pattern": ".+sur_refl_b01.*",
      "nodata": -28672
    },
    "sur_refl_b02": {
      "pattern": ".+sur_refl_b02.*",
      "nodata": -28672
    },
    "sur_refl_b03": {
      "pattern": ".+sur_refl_b03.*",
      "nodata": -28672
    },
    "sur_refl_b04": {
      "pattern": ".+sur_refl_b04.*",
      "nodata": -28672
    },
    "sur_refl_b05": {
      "pattern": ".+sur_refl_b05.*",
      "nodata": -28672
    },
    "sur_refl_b06": {
      "pattern": ".+sur_refl_b06.*",
      "nodata": -28672
    },
    "sur_refl_b07": {
      "pattern": ".+sur_refl_b07.*",
      "nodata": -28672
    },
    "QC_500m": {
      "pattern": ".+sur_refl_qc_500m.*"
    },
    "Day_Of_The_Year": {
      "pattern": ".+sur_refl_day_of_year.*"
    }
  }
}
```

Table attachment 3 – Ingestion file for the MODIS09A1 product, written from scratch to read and interpret the HDF format outputted from the product download from MODIS servers