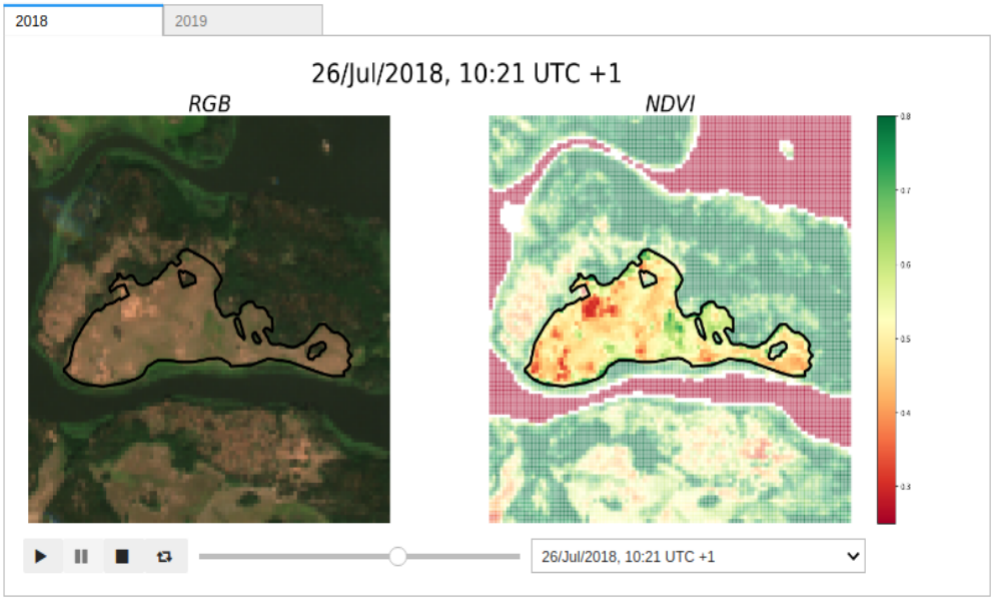
DEGREE PROJECT

# Data Analysis of Earth Observation Data from Copernicus Satellites




Núria Agües Paszkowsky

**Space Engineering, master's level (120 credits)**
**2020**

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering

LULEÅ UNIVERSITY OF TECHNOLOGY

*To my dad...*

# Abstract

Open Data Cubes are platforms that contain open source satellite data and provide analysis tools for governments or organizations. The Swedish version is known as Swedish Space Data Lab (SSDL) and this master thesis was a part of it, providing the first analysis tools of the SSDL. Within a smaller project in the SSDL a drought analysis was done for the region of Mälardalen. The thesis work consisted on developing data analysis methods using packages for machine learning and statistical analysis in Python and Jupyter Notebooks. The drought analysis consisted of a two-year comparison between 2018 and 2019 due to limitations on the data availability. It was found that first year was drier than the second. However, longer time series would be needed in order to observe trends related to possible changes in the climate.

# Acknowledgments

First of all, I would like to thank Tor for giving me the great opportunity of doing this thesis at RISE. Thanks to all my work colleagues for making me feel welcome and specially to Louise and Erik for creating the foosball score board that has united us. Thank you to my supervisor Rickard and my co-worker Johan that have helped me with all the technical problems.

I would also like to thank my family and friends for their constant support. Special mention to Kyriaki for accompanying me through the Kiruna journey and making it feel more bearable. Thanks to Nicolás for encouraging me when I needed it.

Finally, thanks to Maria for her willingness to always help students at campus and Mathias for his guidance.

# Contents

# Figures

# Tables

# Acronyms

**AGDC** Australian Geoscience Data Cube

**AI** Artificial Intelligence

**ALOS** Advanced Land Observing Satellite

**AOT** Aerosol Optical Thickness

**ARD** Analysis Ready Data

**ARD** Automatic Relevance Determinance

**CLD** CLouD Probability Band

**EC** European Commission

**EO** Earth Observation

**ESA** European Space Agency

**EVI** Enhanced Vegetation Index

**FIR** Far Infra Red

**GA** Geoscience Australia

**GIS** Geographic Information System

**GPR** Gaussian Process Regression

**ICE** Infrastructure and Cloud research & test Environment

**IR** Infra Red

**LWIR** Long Wave Infra Red

**MAE** Mean Absolute Error

**MSI** Moisture Stress Index

**MSI** Multi-Spectral Instrument

**MWIR** Medium Wave Infra Red

**MWR** MicroWave Radiometer

**NBR** Normalized Burn Ratio

**NDBI** Normalized Difference Buildup Index

**NDSI** Normalized Difference Snow Index

**NDVI** Normalized Difference Vegetation Index

**NDWI** Normalized Difference Water Index

**NIR** Near Infra Red

**ODC** Open Data Cube

**OLCI** Ocean and Land Colour Image

**POD** Precise Orbit Determination

**RBF** Radial Basis Function

**RGB** Red Green Blue

**RISE** Research Institutes of Sweden

**RMSE** Root Mean Squared Error

**SAR** Synthetic Aperture Radar

**SAVI** Soil Adjusted Vegetation Index

**SCL** Scen CLassification Band

**SDL** Space Data Lab

**SE** Squared Exponential

**SLSTR** Sea and Land Surface Temperature Radiometer

**SNSA** Swedish National Space Agency

**SNW** SNoW Probability Band

**SRAL** Synthetic Aperture Radar ALtimeter

**SSDL** Swedish Space Data Lab

**SWIR** Short Wave Infra Red

**TROPOMI** TROPOspheric Monitoring Instrument

**USGS** United States Geological Survey

**WVP** Scene-average Water VaPour

# Chapter 1

# Introduction

In the past all the data coming from the first satellites were stored in large rolls of tape. In 2011, there was a project called Unlocking the Landsat Archive that had as a main goal copy the Landsat data from tapes onto spinning disks at the National Computational Infrastructure. Geoscience Australia was part of that project and after it, they developed the Australian Geoscience Data Cube (AGDC) which is the origin of the Open Data Cube project (ODC).

As new generations of satellites are being launched, the amounts of stored data keep growing exponentially and therefore appears the necessity of managing that data efficiently. The main objective of the Open Data Cube is to create an open source infrastructure that allows any governments or organizations in the world to easily access, manage and analyze large amounts of Geographic Information System (GIS) data or Earth observation (EO) data.

## 1.1 Motivation

The Swedish Space Data Lab (SSDL) at RISE, in partnership with LTU, Swedish National Space Agency and AI Innovation of Sweden, is a project based on the Open Data Cube. It aims to be a national resource for the Swedish authorities' work on Earth observation data and for the development of AI-based analysis of data generated in space systems. The purpose of the Space Data Lab is to increase the use of data from space for the development of society and industry and for the benefit of the globe. It means support actions to improve sustainability and minimize effect on the environment.

## 1.2 Thesis Aim

This thesis was part of the start up of the Space Data Lab, more specifically, a shorter project within it called Pilot Mälardalen. Its aim was to develop data analysis methods

using packages for machine learning and statistical analysis in Python. The data analysis developed in this master thesis project was a first part of the services that the Space Data Lab will be able to provide.

## 1.3 Objectives

In order to accomplish the thesis aim, the following steps were defined:

1. Become familiar with Python and Jupyter Notebooks.
2. Get comfortable using the Space Data Lab Jupyter Hub.
3. Get started with basic analysis tools taken from the Australian Open Data Cube documentation.
4. Define the use case and select the appropriate spectral index/indices used for analysis.
5. Create a first version of an analysis notebook trying out the first implementation methods.
6. Get familiar with Gaussian Process Regressions and apply it to the analysis.
7. Work on an iterative process until a final version of a notebook is satisfactory.
8. Validate the method.
9. Document all the steps in this report.

## 1.4 Delimitations

The data available in the Space Data Lab at the time this master thesis was being done was crucial in the definition of the use cases. Since this master thesis aim was to develop analysis tools for the Space Data Lab within the Pilot Mälardalen, the use case set the analysis topic of this thesis to drought. The data also had limitations such as having to be rejected in case of clouds, or having a time interval between images dependant on the satellite's trajectory.

## 1.5 Thesis Outline

The following chapter provides the reader with the necessary background to understand this thesis' work. Chapter 3 explains the material and the methods of the thesis. Chapter 4 shows and discusses the obtained results. Chapter 5 summarizes the conclusions and mentions possible future work.

# Chapter 2

# Background

## 2.1 Open Data Cube

The Open Data Cube (ODC) is an open source project that involves the combined actions of people and organizations which aim to build the capability of working with Earth observation data. In other words, it combines the hardware where the data is stored, the software or platform built to access that data, plus the applications or analysis tools with analysis ready data (ARD). This three main parts are clearly represented in Figure 2.1.

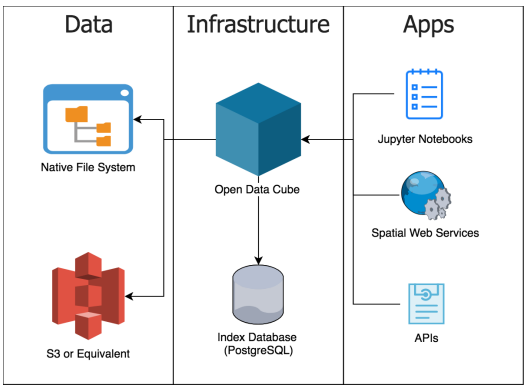



*Figure 2.1: Three Main Parts Composing the Open Data Cube. [1]*

One of the main advantages of the ODC is that it does not require the data to be stored in a specific place or way or that it does not require the user to know where the data is, the Data Cube's software organizes and finds it by itself. Another advantageous

aspect of the ODC is that including the analysis tools makes data available and easy to use for anyone.

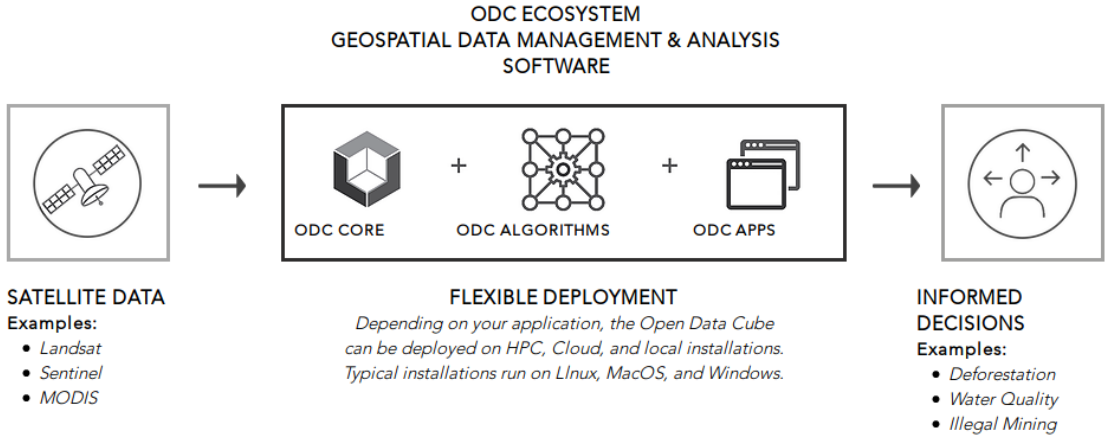



*Figure 2.2: Open Data Cube Ecosystem. [2]*

Figure 2.2 shows the ODC ecosystem. The Landsat, Sentinel and MODIS are the Earth observation satellites whose data is typically in the Data Cube. The ODC consists of a core, algorithms, and applications. On a technical level, it can be separated in data, index, and software. Data is usually file-based, the index allows the user to choose a time and a location and get the data without knowing exactly where the files were stored or how to access them.

The software, which consists of a Python library, allows the user to perform a series of functions for managing data such as indexing data (add records to the index), ingesting data (optimize indexed data for performance) or querying data (returning data in a standard format).

The applications, which will be discussed further in another section, provide enough information to help in decisions in very different areas like environmental, climate change, disaster prevention, security, etc. The ODC will always be 100% open-source.[2]

As mentioned before, the Open Data Cube has its origin in Australia so the first data that were made available in the cube came from the Landsat corresponding to that area. Nevertheless, the objective in 2007 was to achieve a global network of data cubes and to have deployed twenty operational data cubes in twenty countries by the year 2022. Nowadays there are nine operational data cubes in countries like Australia, Colombia, Switzerland, Taiwan, Kenya, Tanzania, Sierra Leone, Ghana, and Senegal. Some others are under development or review as shown in Figure 2.3. The Swedish Space Data Lab is not listed yet as an official ODC.

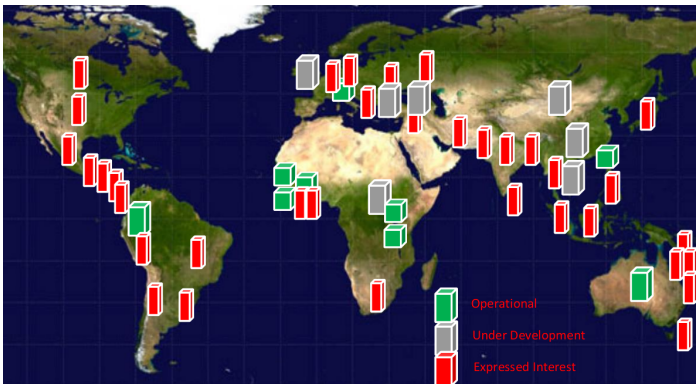



*Figure 2.3: Global Impact of the Open Data Cube. [1]*

With this global network, the impact of satellite data will be increased because the Open Data Cube is a great tool that makes it easier to access, organize and use any layer of data that in the past was not being used. Once the structure is created it provides Analysis Ready Data (ARD) for governments or organizations helping them in decision making for deforestation, water quality, illegal mining or others which is the immediate application of the ODC.[3]

Some examples of the first operational Data Cubes are briefly described below:

### Digital Earth Australia

Digital Earth Australia was the first implementation of a Data Cube and, as said before, it started as a digitization of satellite imagery. It is operative and it is extensively documented through GitHub repositories. They have been the developers of the core functions and basic functionality of the ODC and since it is an open source software, all the other data cubes, including the Swedish, have made use of these functions.

### Africa Regional Data Cube

The aim of this project was to bring data technology to five different African countries that are not typical users of satellite data. Those countries are Ghana, Kenya, Senegal, Sierra Leone, and, Tanzania. It was launched in May 2018 and initially based on Landsat data. Nevertheless, Sentinel 1, Sentinel 2 and ALOS data are planned to be uploaded too. The use cases were agriculture, flooding, urbanization, deforestation, and illegal mining.[2]

#### Swiss Data Cube

The Swiss Data Cube began with 5 years of downloaded Landsat Analysis Ready Data in 2016. Nowadays there are over 30 years of Landsat, Sentinel 1 and Sentinel 2 over the entire country. New data are automatically updated daily as new scenes become available. The Swiss Data Cube's products of interest are urbanization, cloud-free mosaics, and snow cover. It was one of the first adopters of the Data Cube system for a national scale platform.[2]

## 2.2 Satellites Data

The Open Data Cube is based on Earth Observation (EO) data from satellites. The most common ones are the Sentinel and Landsat satellites which will be described in this section.

### 2.2.1 Copernicus Programme and Sentinel Satellites

The Copernicus programme is a partnership between the European Commission (EC) and the European Space Agency and its main goal is to provide Earth observation data that is accurate, timely and easily accessible. The data provided can be used for environmental management, better understanding of the effects of climate change or civil security amongst others. The data offered by Copernicus can provide services or applications mainly in atmosphere, marine, land, climate, emergency and security. [4]

The Sentinel Satellites are a family of satellites specifically designed by the European Space Agency (ESA) to cover the needs of the Copernicus programme. The missions that have already been sent are Sentinel 1, 2, 3 and 5P. Sentinel 4, 5 and 6 are under development. [5]

#### Sentinel 1

The Sentinel-1A was launched in April 2014 and the Sentinel-1B in April 2016. Together they form a constellation and take part in the same mission. The expected life of the mission is for at least seven years.

The main objective of the mission is to provide radar data independently of the weather conditions, day or night. It carries an instrument called C-band Synthetic Aperture Radar which operates at wavelengths that are not hampered by clouds or lack of light. The main applications of the mission are such as marine, land and disaster monitoring. [6]

### Sentinel 2

The Sentinel-2A was launched in June 2015 and the Sentinel-2B in March 2017. Mostly all the Sentinels come in pairs, they form a constellation of two satellites with a phase of 180° like seen in Figure 2.4. The expected life of the mission is for at least seven years.

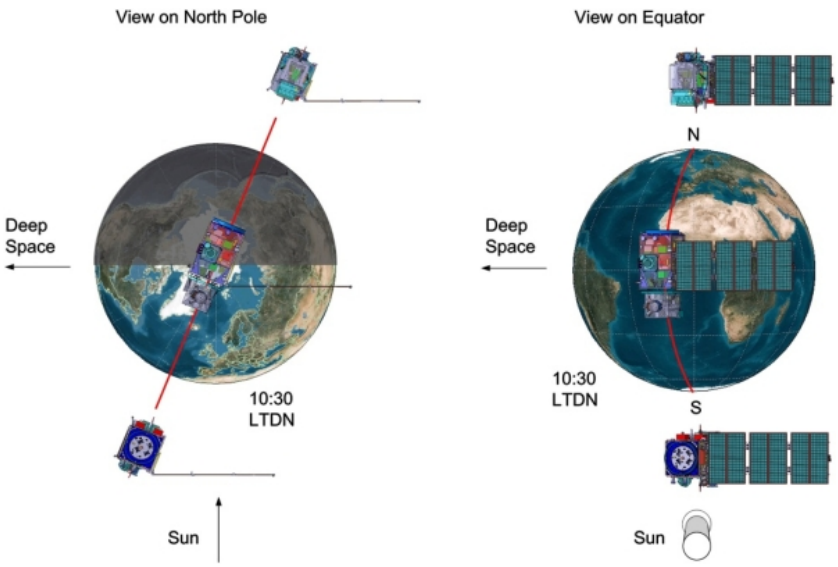



*Figure 2.4: Orbital Configuration of the Sentinel-2 Constellation. [6]*

The main objective of the mission is to provide multi-spectral images of the Earth and continue the mission of the series of satellites SPOT and the USGS Landsat Thematic Mapper instrument. The data collected can be used for different applications such as climate change, land monitoring, security or emergency management. The data of the Sentinel-2 was used for this master thesis. [6]

The instrument on-board the Sentinel-2 is the Multi-Spectral Instrument (MSI) which has thirteen spectral bands or channels in the visible, near infrared and short wave infrared spectral ranges. Figure 2.5 is a screenshot of a table that summarizes the specifications.

| Spatial Resolution (m) | Band Number | S2A | | S2B | |
|---|---|---|---|---|---|
| | | Central Wavelength (nm) | Bandwidth (nm) | Central Wavelength (nm) | Bandwidth (nm) |
| 10 | 2 | 492.4 | 66 | 492.1 | 66 |
| | 3 | 559.8 | 36 | 559.0 | 36 |
| | 4 | 664.6 | 31 | 664.9 | 31 |
| | 8 | 832.8 | 106 | 832.9 | 106 |
| 20 | 5 | 704.1 | 15 | 703.8 | 16 |
| | 6 | 740.5 | 15 | 739.1 | 15 |
| | 7 | 782.8 | 20 | 779.7 | 20 |
| | 8a | 864.7 | 21 | 864.0 | 22 |
| | 11 | 1613.7 | 91 | 1610.4 | 94 |
| | 12 | 2202.4 | 175 | 2185.7 | 185 |
| 60 | 1 | 442.7 | 21 | 442.2 | 21 |
| | 9 | 945.1 | 20 | 943.2 | 21 |
| | 10 | 1373.5 | 31 | 1376.9 | 30 |

*Figure 2.5: Wavelengths and Bandwidths of the MSI Instrument. [6]*

The data produced by the Sentinel-2 can be divided in different levels depending on the "rawness" of the data. It is possible to download Level-1C and 2A from the ESA website and that was done at RISE in order to ingest data into the Swedish Space Data Lab.

Level-1C data corresponds to top of atmosphere reflectances whereas Level-2A is one step further processed and corresponds to bottom of atmosphere reflectances. The level-2A data were the one used in this master thesis. ESA performs some radiometric and geometric corrections to the 2A level data. Furthermore, this level comes with five extra computed bands:

- Aerosol Optical Thickness (AOT).
- Cloud Probability Band (CLD).
- Scene Classification Band (SCL). This was a crucial band for this thesis' work and it is explained in detail in Section 3.3.4.
- Snow Probability Band (SNW).
- Scene-average Water Vapour (WVP).

### Sentinel 3

The Sentinel-3A was launched in February 2016 and the Sentinel-3B in April 2018. Together they form a constellation and take part in the same mission. The expected life of the mission is for at least seven years.

The main objective of the mission is to provide measurements such as land and sea surface temperatures in order to improve environmental and climate change monitoring. It carries five instruments: Ocean and Land Colour Image (OLCI), Sea and Land Surface Temperature Radiometer (SLSTR), Synthetic Aperture Radar (SAR) Altimeter (SRAL), Microwave Radiometer (MWR) and Precise Orbit Determination (POD) instruments. [6]

### Sentinel 5P

The Sentinel-5P is the precursor of the Sentinel-5 mission. It was launched in October 2017 to fill in the gap between the satellites Envisat and Sentinels 4 and 5. Its objective is to monitor the Earth's atmosphere and measure air quality, ozone and UV radiation. It carries the instrument TROPOMI which is the acronym for TROPOspheric Monitoring Instrument. [6]

## 2.2.2 Landsat Satellites

Landsat satellites were not used for this master thesis but the Open Data Cube started with the purpose of making Landsat data available for everybody. The Landsat satellites have the advantage of covering long timeline series as seen in Figure 2.6. The first Landsat was launched in 1972 and since then there have always been at least two operating Landsat satellites orbiting Earth.

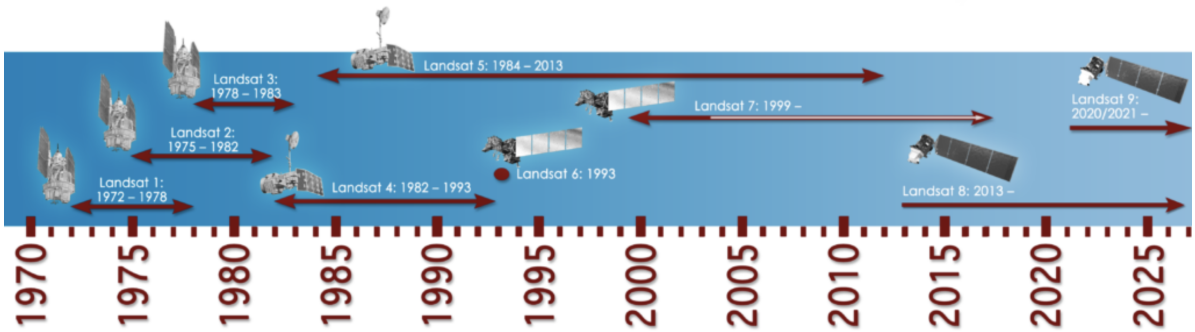



*Figure 2.6: Timeline of the Landsat Satellites. [7]*

## 2.3 Swedish Space Data Lab

The Swedish Space Data Lab is a two-year project funded by Vinnova that started in June 2019. It is done as a collaboration between four partners: RISE (Research Institutes of Sweden), the Swedish National Space Agency (SNSA), AI Innovation of Sweden and Luleå Tekniska Universitet (LTU).

Inspired by the Open Data Cubes, the Swedish Space Data Lab aims to become the Swedish Data Cube. It has data geographically located in Sweden and its main purpose is to boost the usage of space data that helps in development of society and industry. It aims to be the main data hub the Swedish authorities seek for earth observation data.

The end users can be very different, it could go from public authorities to large companies or even private persons or individual farmers. The Space Data Lab uses and develops AI-based analysis data in order to solve civil, environmental, agricultural, and other type of problems. [8]

During the time this master thesis was being done, the Space Data Lab was just starting and at the same time its platform was being built and developed, there was a minor project inside called Pilot Mälardalen. The Pilot Mälardalen was a proof of concept and it was ongoing between September and December 2019. The end user was Landstyrelsen Västmanland (County Administrative Board of Västmanland) and the main focus was the study of drought in that area. By the time this study was done, the only available data within the SDL was Sentinel-2 level 2A from 2018 to date.


*Figure 2.7: First Area of Sentinel-2 Data Ingested in the SSDL.*

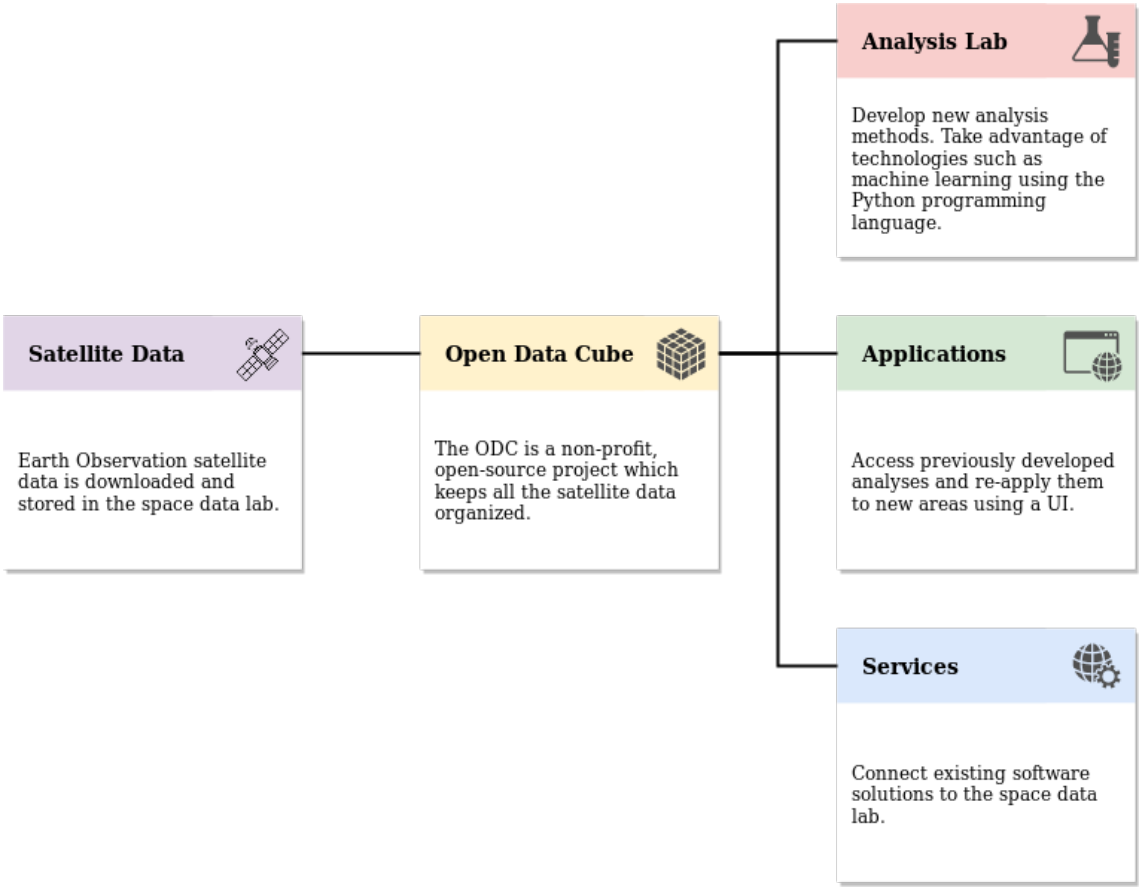



*Figure 2.8: Swedish Space Data Lab Schematic Overview. [8]*

Figure 2.8 shows a schematic overview of the Space Data Lab structure. First the satellite data is downloaded and stored in the ICE servers. Then the Open Data Cube (ODC) structure is built around these data by indexing, ingesting and organizing the data.

The three last boxes at the right show the three possible outcomes from the Open Data Cube. Analysis Lab would be for example the use of Jupyter Notebooks. The box for Applications refers to any API that could be built in a website, for example a snow analysis, and that website has an interface and allows the user to select an area, then the API access the data and the analysis tools from the ODC to display the results in the website. The last box is about offering the possibility to combine other software with the DC, for example Arc-GIS applications.

Since the Space Data Lab was in an early stage at the time of this master thesis, the applications and services had not been developed yet.

## 2.4 Data Cube Applications

A data cube is designed to be able to analyze scenario data together with historical data. It provides the conditions for developing functions and services that can be used to identify, illustrate and evaluate climate risks. Furthermore, evaluation of the benefits of planned measures is possible, but also calculation of costs when these have a spatial and timely aspect.

The ODC supports a broad range of applications including land, water, cloud, and time series analysis. Some example applications are cloud-free mosaics or the calculation of spectral indices.

### 2.4.1 Spectral Indices

Spectral indices calculate the relative magnitudes of wavelength components in order to indicate the relative abundance of any feature of interest. The most popular ones are vegetation indices but there are also other indices for burned areas, water, geologic or man-made features. The two indices used in this master thesis were NDVI and MSI. Some other indices are briefly mentioned in the report.

All these indices are defined using the names of the electromagnetic spectrum within the visible and infrared (IR) ranges. In the case of the visible range names of colours are used. The IR is divided as shown in Table 2.1:

| **Region** | **Abreviation** | **Typical wavelengths ($\mu$m)** |
|---|---|---|
| Near Infra Red | NIR | [0.75 - 1.4] |
| Short Wave Infra Red | SWIR | [1.4 - 3] |
| Medium Wave Infra Red | MWIR | [3 - 8] |
| Long Wave Infra Red | LWIR | [8 - 15] |
| Far Infra Red | FIR | [15 - 1000] |

*Table 2.1: Infra Red Subregions.*

#### NDVI

NDVI stands for Normalized Difference Vegetation Index. It tells whether there is green vegetation or not in the pixel or area that it is being studied. Its physical principle is based on the pigment in the leaves strongly absorbing visible light from 0.4 to 0.7$\mu$m which use it for the photosynthesis. On the other hand, the leaves strongly reflect near-infrared light from 0.7 to 1.1$\mu$m. The number of leaves that a plant has affects how much these wavelengths are affected. The NDVI index can be calculated by:

$$NDVI = \frac{NIR - Red}{NIR + Red} \tag{2.1}$$

Figure 2.9 shows the difference in reflectances between dead, stressed or healthy leaves. The biggest difference is between Red and NIR and that's the reason those bands were chosen for the NDVI. Healthy leaves reflect much more at NIR than stressed or dead leaves whereas the red reflectance is lower in healthy than unhealthy. This makes a greater difference between them and explains the higher values of NDVI for healthy vegetation. Figure 2.10 shows the difference between the spectral signatures of healthy and unhealthy vegetation.


*Figure 2.9: Healty and Unhealthy Vegetation Reflectance.[9]*

The values of NDVI can vary from [-1, 1]. If the value is in the range of [-1, 0] it represents water bodies, [-0.1, 0.1] are usually rocks, sand or snow, [0.2, 0.5] represent grasslands or crops, and [0.6, 1.0] correspond with dense vegetation or tropical rain-forest. The higher the value of NDVI is, the higher the reflection of near-infrared is which means more green vegetation.

### MSI

MSI stands for Moisture Stress Index and as its name indicates it measures leaf water content using reflectances. It is defined as the ratio of the bands:

$$MSI = \left[\frac{1600nm}{820nm}\right] \tag{2.2}$$

Since it is a ratio and it is not based on a difference between two bands like NDVI, one of the bands has to be used as reference. That is the 820nm band because water content

does not affect, or affects very little, the reflectance measured at that wavelength. The 1600nm band can see different reflectances in spectral signatures depending on the water content, therefore it can characterize water stress.

Some studies to find the most suitable bands to characterize MSI were done, [10], and five different wavelengths were tested: 560nm, 680nm, 800nm, 1440nm and 1600nm. The results found the best ratios to be 1440nm and 1600nm divided by any of the other three wavelengths, being 1440nm and 1600nm the useful wavelengths for leaf water content characterization. However, the absorption band of atmospheric water vapour is around 1440nm which would give a low signal to noise ratio for that wavelength. Hence, the optimal ratio for space applications is R(1600nm)/R(800nm).

Figure 2.10 shows the differences in spectral signatures of healthy and dry vegetation in green and red respectively. It can be seen that around 800nm the difference between the signatures is minimum. On the other hand, around 1400nm and 1600nm the difference is larger.

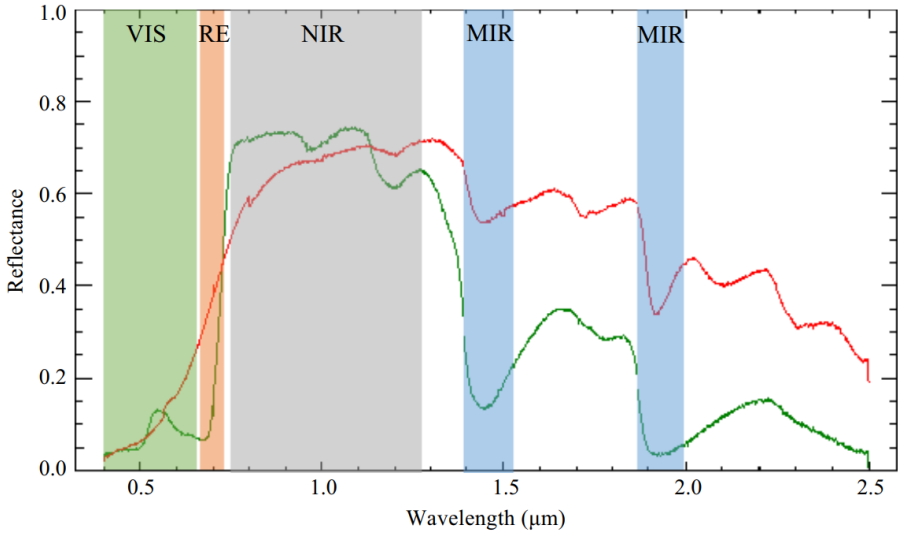



*Figure 2.10: Healthy and Dry Vegetation Spectral Signatures in Green and Red, respectively. [11]*

The MSI has applications such as stress analysis, productivity prediction and modelling, fire hazard condition analysis or studies of ecosystem physiology. Its values can range from 0 to more than 3 being higher values associated to higher water stress. This is usually inverted for other water vegetation indices for which higher values indicate healthier vegetation. [12]

**Other Indices**

Some other spectral indices that could be calculated using the Sentinel-2 bands are:

- **NBR.** Normalized Burn Ratio, used to estimate burn severity. It can be calculated according to the formula below. Like the NDVI index, it can have a range in between [-1, 1]. For values smaller than -0.1, it indicates post fire regrowth. Values in between [-0.1, +0.1] mean unburned vegetation, [0.1, 0.27] low-severity burn, [0.27, 0.66] moderate severity burn and greater than 0.66 indicate high severity burn.
$$NBR = \frac{NIR - SWIR_1}{NIR + SWIR_1} \tag{2.3}$$
- **NDBI.** It stands for Normalized Difference Buildup Index and it is used for the detection of urban areas. It is similar to NBR but it has opposite sign.
$$NDBI = \frac{SWIR_1 - NIR}{SWIR_1 + NIR} \tag{2.4}$$
- **NDWI.** It stands for Normalized Difference Water Index and it is used for water detection.
$$NDWI = \frac{Green - NIR}{Green + NIR} \tag{2.5}$$
- **NDSI.** It is a modified NDWI used for snow or water detection. It stands for Normalized Difference Snow Index.
$$NDSI = \frac{Green - SWIR_1}{Green + SWIR_1} \tag{2.6}$$
- **SAVI.** It is the Soil Adjusted Vegetation Index used for soil response.
$$SAVI = \frac{NIR - Red}{NIR + Red + 0.5} \cdot 1.5 \tag{2.7}$$
- **EVI.** Enhanced Vegetation Index it is another vegetation index. It enhances the vegetation signal with improved sensitivity in high biomass regions.
$$EVI = 2.5 \cdot \frac{NIR - Red}{NIR + 6 \cdot Red - 7.5 \cdot Blue + 1} \tag{2.8}$$

## 2.5 Gaussian Process Regression

### 2.5.1 GPR Background

Supervised learning is the machine learning task that learns input-output mapping based on empirical data. If the output is continuous then the problem is called regression whereas if the output is discrete it is called classification.

In general there is an input vector $\mathbf{x}$ and an output or target $y$. The vector $\mathbf{x}$ has the information of all the input variables that can be multiple. Combining the input and the output of the $n$ observations, a dataset $\mathcal{D}$ can be defined as $\mathcal{D} = \{(x_i, y_i) \mid i = 1, ..., n\}$.

The main goal of a supervised learning problem is to predict for new inputs $\boldsymbol{x_*}$ that are not part of the training dataset $\mathcal{D}$. In order to do so, it is necessary to make some assumptions of the function $f$ that makes predictions for every possible input. There are many different approaches but the main focus for this master thesis was on the so called Gaussian Process Regressions. [13]

### Gaussian Distribution

A Gaussian process is basically a generalization of the Gaussian probability distribution and regression, as mentioned above, has to do with the continuity of the output. The probability density function of a variable is Gaussian if it follows the formula:

$$\varphi(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Big(\frac{-(x-m)^2}{2\sigma^2}\Big), \quad x \in \mathbb{R} \tag{2.9}$$

for $m \in \mathbb{R}$ and $\sigma > 0$. Every Gaussian distribution is well defined by its mean $m$ and its standard deviation $\sigma$ so it can be written as $X \sim \mathcal{N}(m, \sigma^2)$. This corresponds to a uni-dimensional case (1D) but it can be generalized for multiple dimensions and then it is called multivariate Gaussian distribution. Then, it follows the probability density function below:

$$\varphi(x) = (2\pi)^{-D/2}|K|^{-1/2} \exp\Big(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m})^T K^{-1}(\boldsymbol{x}-\boldsymbol{m})\Big), \quad x \in \mathbb{R}^D \tag{2.10}$$

being $m$ its mean and $K$ the covariance matrix. The mean vector $\mathbf{m} \in \mathbb{R}^D$ has to be such that $m_i = \mathbb{E}[X_i]$ and the covariance matrix K is symmetric positive semi-definite $K \in \mathbb{R}^{D\times D}$ and has the elements $K_{ij} = Cov(X_i, X_j) = \mathbb{E}[(X_i - m_i)(X_j - m_j)]$. A multivariate Gaussian distribution can then be represented as $X \sim \mathcal{N}(m, K)$ or $X \sim \mathcal{N}_{\mathcal{D}}(m, K)$ needing only its mean and covariance matrix. [14]

### Bayes' Theorem

As mentioned before a Gaussian process is a generalization of the Gaussian probability distribution that has just been described. The Gaussian Process works by assigning a prior probability to every possible function that can predict the output, assigning higher probabilities to the functions considered to be more likely. In order to make predictions it uses Bayesian methods. The well known Bayes' rule states that: [13]

$$posterior = \frac{likelihood \times prior}{marginal\ likelihood}, \quad p(w \,|\, y, X) = \frac{p(y \,|\, X, w) \cdot p(w)}{p(y \,|\, X)} \tag{2.11}$$

The interpretation for Equation 2.11 is that a posterior distribution, $p(w \,|\, y, X)$, can be calculated by using the information on the observed data, $p(y \,|\, X, w)$, and assigning a prior distribution to a parameter $w$, $p(w)$. Therefore, a posterior distribution is updated or relocated based on the evidence (data) and a prior assumption on the distribution.

### Predictive Distribution

Using Bayes Law, a posterior distribution is calculated and then used to get predictions at the mentioned points, $\boldsymbol{x}_*$, that are not part of the initial dataset. The difference between $\mathbf{x}$ and $\boldsymbol{x}_*$ is that the former refers to the input points which are all part of the dataset. The latter refers to new points instead. The predictive distribution uses the average over all possible parameter values and weights it by their posterior probability:

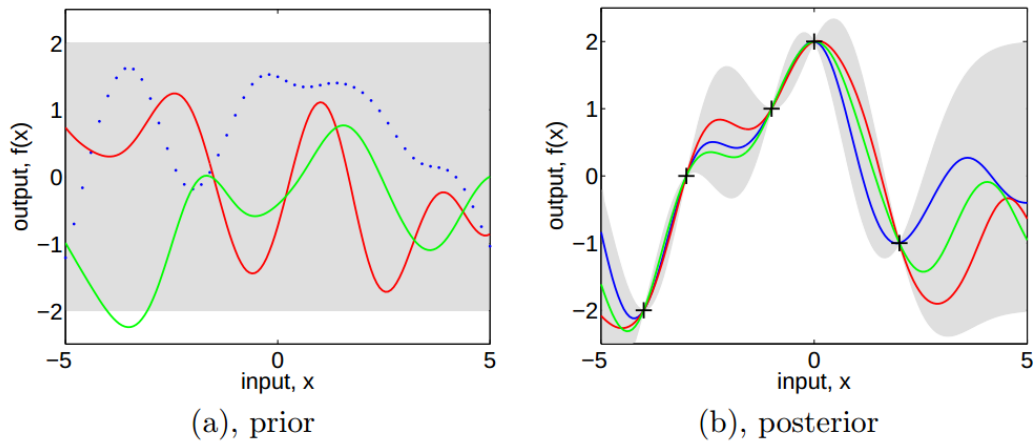
$$p(f^* \,|\, x^*, y, X) = \int_w p(f^* \,|\, x^*, w)\, p(w \,|\, y, X)\, dw \tag{2.12}$$

Usually the prior and likelihoods are assumed to be Gaussian and even though it makes things simpler, that is not the reason why it is chosen like that. The real reason is because it is actually quite accurate with the reality. The predictive distribution also follows a Gaussian distribution providing a mean value for the predictions with an uncertainty band obtained from its variance. [13]

### Prior and Posterior Interpretation

A good way to understand the prior and posterior concepts is by visual representation as shown in Figure 2.11. Its subfigure (a) is a plot of three priors which corresponds to assumptions made before having any data points or information, therefore the three functions are plotted randomly.

The subfigure (b) shows how the three random functions are updated at the posterior based on the information obtained from five observations. The shaded area represents the confidence band or uncertainty which is obtained using the standard deviation. The mean $\pm$ two times the standard deviation corresponds to a 95% confidence region. [13]




(a), prior (b), posterior

*Figure 2.11: Examples of Prior and Posterior for a Gaussian Process. [13]*

### 2.5.2 GPR Theoretical Overview

The best interpretation for a Gaussian Process when used for Regression is a Gaussian distribution over functions, and as a Gaussian, it can be specified by its mean, *m(x)*, and covariance functions, *k(x, x')*: [14]

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \tag{2.13}$$

A training dataset was defined earlier as $\mathcal{D} = \{(x_i, y_i) \mid i = 1, ..., n\}$ and that is equivalent to $\mathcal{D} = (X, \mathbf{y})$ being $X$ a design matrix that contains all the input vectors $X = (x_1, x_2, ...x_n)$ and $\boldsymbol{y}$ the target vector $\mathbf{y} = (y_1, y_2, ...y_n)^T$.

The covariance matrix is interpreted as the covariances between all pair of points in two datasets $X^1$ and $X^2$, $K(X^1, X^2)$, where $K(X^1, X^2)_{ij} = k(x_i^1, x_j^2)$. The mean, *m(x)*, is usually assumed to be zero for simplicity.

It is necessary to define a model that can connect the output *y(x)* to the Gaussian process function *f(x)*. Normally, the output values do not correspond to the values of $f$ due to the presence of noise. The usual way of modelling it is by adding a Gaussian noise, hence $y(x) = f(x) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$.

The covariance between the outputs is written as:

$$Cov(y_p, y_q) = k(x_p, x_q) + \sigma_n^2 \delta_{pq} \quad or \quad Cov(y) = K(X, X) + \sigma_n^2 I \tag{2.14}$$

The joint distribution of both, the observed values for the outputs $y$ and the test input points $f_*$ is:

$$\begin{pmatrix} y \\ f_* \end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} K(X, X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{pmatrix}\right) \tag{2.15}$$

The predictive distribution can be expressed for both, the Gaussian process regression $f_*$ and for the test targets $y_*$. As defined in 2.11, the distribution of the test input points conditioned on the training dataset is:

$$f_* \,|\, X^*, X, y \sim \mathcal{N}(\mathbb{E}[f_* \,|\, X^*, X, y], Cov(f_* \,|\, X^*, X, y)) \tag{2.16}$$

$$\mathbb{E}[f_* \,|\, X^*, X, y] = K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} y \tag{2.17}$$

$$Cov(f_* \,|\, X^*, X, y) = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} K(X, X_*) \tag{2.18}$$

In the same way, the predictive distribution for the test targets $y_*$ is calculated by adding the Gaussian noise $\sigma_n^2 I$ to the covariance $Cov(f_* \,|\, X^*, X, y)$, leaving: [14]

$$y_* \,|\, X^*, X, y \sim \mathcal{N}(\mathbb{E}[y_* \,|\, X^*, X, y], Cov(y_* \,|\, X^*, X, y)) \tag{2.19}$$

$$\mathbb{E}[y_* \,|\, X^*, X, y] = K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} y \tag{2.20}$$

$$Cov(y_* \,|\, X^*, X, y) = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} K(X, X_*) + \sigma_n^2 I \tag{2.21}$$

### 2.5.3 GPR Python Implementation

There are many libraries in Python to implement any Gaussian Process Regression, and the one chosen for this master thesis was GPflow [15]. The three most important choices during model selection are the mean function, the covariance function kernel and the hyperparameters.

#### Mean Function

The mean function is usually zero or the mean of the training data set. Its importance lies with the fact that when two data points are quite separated, the prediction function will tend to that mean value.

#### Covariance Function Kernel

When implementing a GPR, talking about covariance function or kernel is the same. Its choice is the most important one because it is the covariance function that describes the function to be predicted. In other words, it describes the relationship between the data points and the predictions.

The notion of similarity between data points is implied within the covariance function. As a common assumption, it is accepted that points with similar inputs will have also similar output values. Therefore, a prediction near a data point should have an outcome very similar to the one in the data point. [14]

There are many different types of covariance functions such as constant, linear, polynomial, squared exponential, Matérn, exponential, $\gamma$-exponential, rational quadratic, neural networks, etc. According to Rasmussen and Williams the squared exponential (SE) is the most commonly used and it is the one that was used in this master thesis. The equation that defines a SE is as follows:

$$k_y(x_p, x_q) = \sigma_f^2 \exp\Big( - \frac{1}{2\ell^2}(x_p - x_q)^2 \Big) + \sigma_n^2 \delta_{pq} \tag{2.22}$$

being $\sigma_f^2$ the signal variance, $\ell$ the length-scale and $\sigma_n^2$ the noise variance. All three parameters are also called hyperparameters.

### Hyperparameters

The values of the hyperparameters can be trained and therefore obtain the optimal values by maximizing the log marginal likelihood. Nevertheless, it is also possible to pre-define their values and condition the prediction function. As Figure 2.12 shows, depending on the values assigned to $\sigma_f^2$, $\ell$ and $\sigma_n^2$, the mean and the confidence band have different outcomes.

The effect of length parameter $\ell$ is compared between the two subplots of the first row of Figure 2.12. Higher values of $\ell$ give a smoother function because it means that the predicted points have a relevant correlation to the closest data points along a particular axis for longer and abrupt changes are less common. When the values of the length-scale are low, the predicted points become quickly uncorrelated to the data points and that causes the mean predicted function to look more teetering and the confidence bands between data points to increase rapidly.

The signal variance $\sigma_f^2$ has an influence on the vertical variation of the function. The second row of Figure 2.12 illustrates the differences, for high values of $\sigma_f^2$ the confidence band becomes much wider outside the training data region.

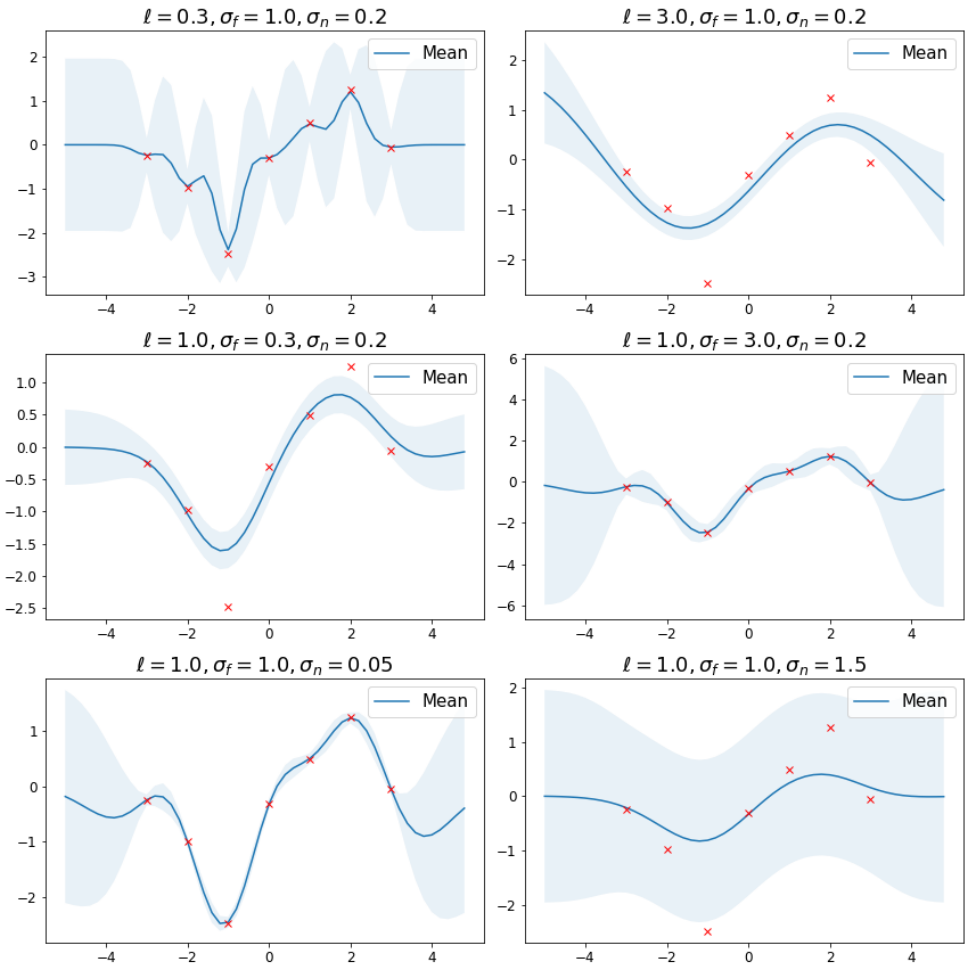



*Figure 2.12: Effects of the Hyperparameters.*

The noise variance $\sigma_n^2$ corresponds to the amount of noise when realizing the observations. It does not have to be modelled if the noise is not being considered or does not exist. The last row of Figure 2.12 exemplifies its effect, when the noise is high the approximation becomes coarser and the confidence band wide in order to avoid over-fitting. In the contrary, if there is no noise, the confidence band becomes very narrow. [13]

### 2.5.4 Advantages and Disadvantages

Gaussian Processes have the advantages:

- It is easy to define confidence intervals and evaluate if the fitting is good enough thanks to the use of Gaussian probabilities.

- Versatility in kernel specification. There are pre-defined kernels but it is always possible to create and customize new ones.
- The observations are interpolated in prediction.

On the other hand, the main disadvantages of Gaussian processes are:

- Not being sparse which means that they use the full sample to predict.
- Not efficient in high dimensional spaces or big datasets. [13]

## 2.6 Validation of the Method

In order to validate the GPR done in this master thesis, a method called cross-validation was used. It consisted of dividing the dataset into two: training and validation. Then the predicted values were compared to the real values using the Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE).

### 2.6.1 Root Mean Squared Error

The RMSE is used to measure the error of a model that predicts quantitative data. It is formally defined as: [16]

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \tag{2.23}$$

being $n$ the number of observations, $\hat{y}_i$ the predicted values and $y_i$ the observed values in the original dataset.

### 2.6.2 Mean Absolute Error

The MAE also measures the error of a model that predicts data but it has an advantage compared to the RMSE. It is better at ignoring the effects of possible outliers, giving a more reasonable error. The formal definition is as follows: [16]

$$MAE = \frac{1}{n} \sum_{i=1}^{n} \mid \hat{y}_i - y_i \mid \tag{2.24}$$

being $n$ the number of observations, $\hat{y}_i$ the predicted values and $y_i$ the observed values in the original dataset.

# Chapter 3

# Material & Method

## 3.1 Software

This subsection describes the software used, directly or indirectly, for this master thesis.

### 3.1.1 GitHub

GitHub is an online platform used for developing software. There, many developers store their code in repositories and it is free to use and access by anyone. It is a perfect tool for collaborations because it allows version control, bug tracking, etc. All the documentation relating the Open Data Cube used for this master thesis can be found in a GitHub repository.

### 3.1.2 Kubernetes Platform

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications. The Space Data Lab requires substantial processing and power unit for large scale image retrieval and analysis. In order to provide enough computational power, a platform is deployed on the ICE cluster at RISE. Kubernetes manages all the resources, scales the system and makes sure that GPU, CPU, memory and storage are efficiently shared in a multi-user cluster.

### 3.1.3 Jupyter Notebooks

The main software used for this thesis is Jupyter Notebooks which is an open-source web application that makes possible to create documents combining equations, interactive

code, visualizations and markdown text. It is a Python language based software. Figure 3.1 shows what the environment looked like.

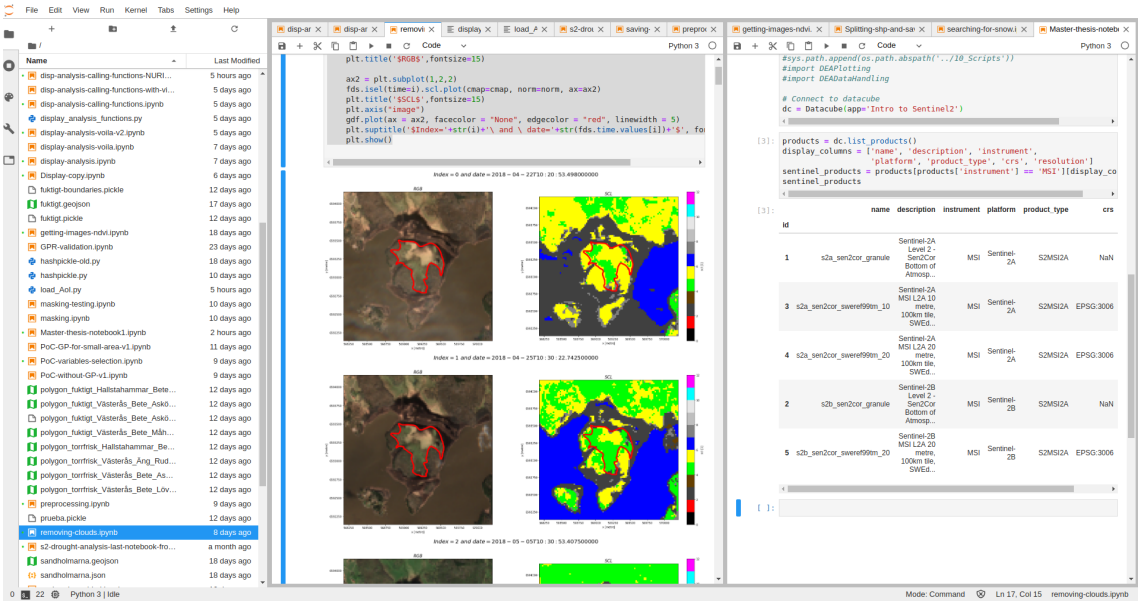


*Figure 3.1: Jupyter Notebooks in JupyterLab Environment.*

### 3.1.4 Software Python Libraries

Python is an open-source programming language designed with the purpose of being easy to read and implement. It is well known that it does not come with every library implemented, but since it is open source, anyone can develop a library that can then be imported to Python and used by everyone. During this master thesis many libraries have been used, some common libraries like pandas or numpy, but also others not so common like the datacube. The most important ones are briefly summarized below:

#### Datacube

The Open Data Cube Core library was developed by Geoscience Australia and other institutional partners. It basically consists of a software that contains functions to communicate with a database which contains information about the satellite data. The possibilities that this library offers are very wide, amongst others, categorize huge amounts of Earth Observation data, provide a Python based API to access that data or provide Exploratory Data Analysis tools.

### Pandas

Pandas is a library used for data manipulation and analysis. It allows the user to manipulate data in the form of tables or time series. It is good when the data structures have two dimensions or less.

### Xarray

Xarray is based on pandas and numpy. It is used for working with labeled N-dimensional arrays. Even though numpy is the usual library for working with ND arrays, xarray has some advantages that make easier working with that type of arrays. For example, xarray has named dimensions instead of axis labels, also NaN or heterogeneous data can be easily handled with xarray.

### Matplotlib

Matplotlib is the commonly used plotting library for Python. It is used for 2D plotting or to export images into other environments.

### Geopandas

Geopandas is a library for working with geospatial data. It is based on pandas and shapely, it extends the pandas' datatypes to enable spatial operations on geometric objects. Those operations are performed by the library shapely, specialized on manipulation and analysis of planar geometric objects.

### Rasterio

Rasterio is the library made to read and write Geographic Information Systems (GIS) files like satellite images or terrain models. Rasterio converts them into numpy N-dimensional arrays and GeoJSON.

### Ipywidgets

Ipywidgets is a library that allows the creation of interactive widgets in the Jupyter Notebooks or IPython kernel. There are many different types such as buttons, slide bars, date pickers, etc. This makes interaction between the data and the user much more immersive.

#### GPflow

It is an open source library for building Gaussian process models. It is based on TensorFlow and has its origins in GPy. It has all the basic models of GPR implemented so they can be used straightforward. It has less code than GPy because TensorFlow is in charge of all the heavy computation. TensorFlow is a powerful numerical package very popular for deep learning. [15]

#### Pickle

Pickle is a module in Python that can be used to serialize/de-serialize objects. "Pickling" an object means to convert it into a byte stream and "unpickling" is the reverse action.

## 3.2 Hardware

### 3.2.1 GPU

All the data from the Space Data Lab is stored in the ICE datacenter facilities of RISE. At the time of the work for this master thesis ten Dell Power Edge R730 equipped with GPU were used in the cluster were the JupyterLab platform was deployed.

## 3.3 Jupyter Notebooks Preprocessing

After having defined all the material used for this master thesis in the two previous subsections, the following sections will describe the method. The use case for the first pilot project in the Space Data Lab was defined to be drought and the two indices established to study it were the Normalized Difference Vegetation Index (NDVI) and the Moisture Stress Index (MSI). Both indices have been previously defined in Section 2.4.1. The steps that were taken are explained in the following subsections.

### 3.3.1 Import the Datacube and Check the Current Available Products

As mentioned before the datacube is a library in Python that can be imported. The list of products available when this thesis was being done consisted of five products as listed in Table 3.1:

| ID | Name | CRS | Resolution |
|---|---|---|---|
| 1 | s2a_sen2cor_granule | EPSG:4326 | 10, 20, 60 |
| 2 | s2b_sen2cor_granule | EPSG:4326 | 10, 20, 60 |
| 3 | s2a_sen2cor_sweref99tm_20 | EPSG:3006 | 20 |
| 4 | s2b_sen2cor_sweref99tm_20 | EPSG:3006 | 20 |
| 5 | s2a_sen2cor_sweref99tm_10 | EPSG:3006 | 10 |

*Table 3.1: Available Products in the Swedish Space Data Lab to Date.*

Products 1 and 2 correspond to Sentinel-2A and 2B respectively. They are the first ones that were uploaded and indexed in the Swedish Space Data Lab and use a Coordinates Reference System (CRS) of EPSG:4326 which correspond to latitude/longitude coordinates. The products that they offer are level-2A products, therefore all 12 bands can be found repeated as many times as available resolutions (10m, 20m and 60m). In addition, there are six new bands named Aerosol Optical Thickness (AOT), Cloud Probability (CLDPRB), Scene Classification Layer (SCL), Snow Probability (SNWPRB), Water Vapour (WVP) and True Color Image (TCI).

Products 3, 4 and 5 use EPSG:3006 which corresponds to sweref99tm, a projected coordinate system for areas in Sweden. They use only 20m resolution or 10m in the case of product 5. Not all the bands have been uploaded. Products 4 and 5 have the following bands: blue, green, red, red_edge_1, red_edge_2, red_edge_3, nir_2, swir_2, swir_3, aot, scl, tci and wvp. Product 5 has only 10m resolution bands and those bands are: blue, green, red, nir_1, aot, tci and wvp.

The products used for the analysis are 4 and 5 combined because Sentinel-2A and 2B are a constellation of satellites, therefore they have a phase between each other of 180°. The combination of both products allow the user to have more dates available.

### 3.3.2 Areas of Interest

During the development of the Pilot Mälardalen seven different areas in the region of Västmanland were provided as shapefiles. A shapefile is a geospatial vector data format for geographic information system software. Those were the areas of interest were the analysis had to be done. The pre-processing and analysis worked independently of the chosen area. The pre-processing steps will be explained using only one area at a time for visualization examples.

Figure 3.2 is an overview map containing all the seven areas which were classified as humid or dry healthy, from now on named as "fuktigt" and "torrfrisk" respectively. Three red squares were drawn in Figure 3.2 in order to zoom in and identify all the areas of interest.

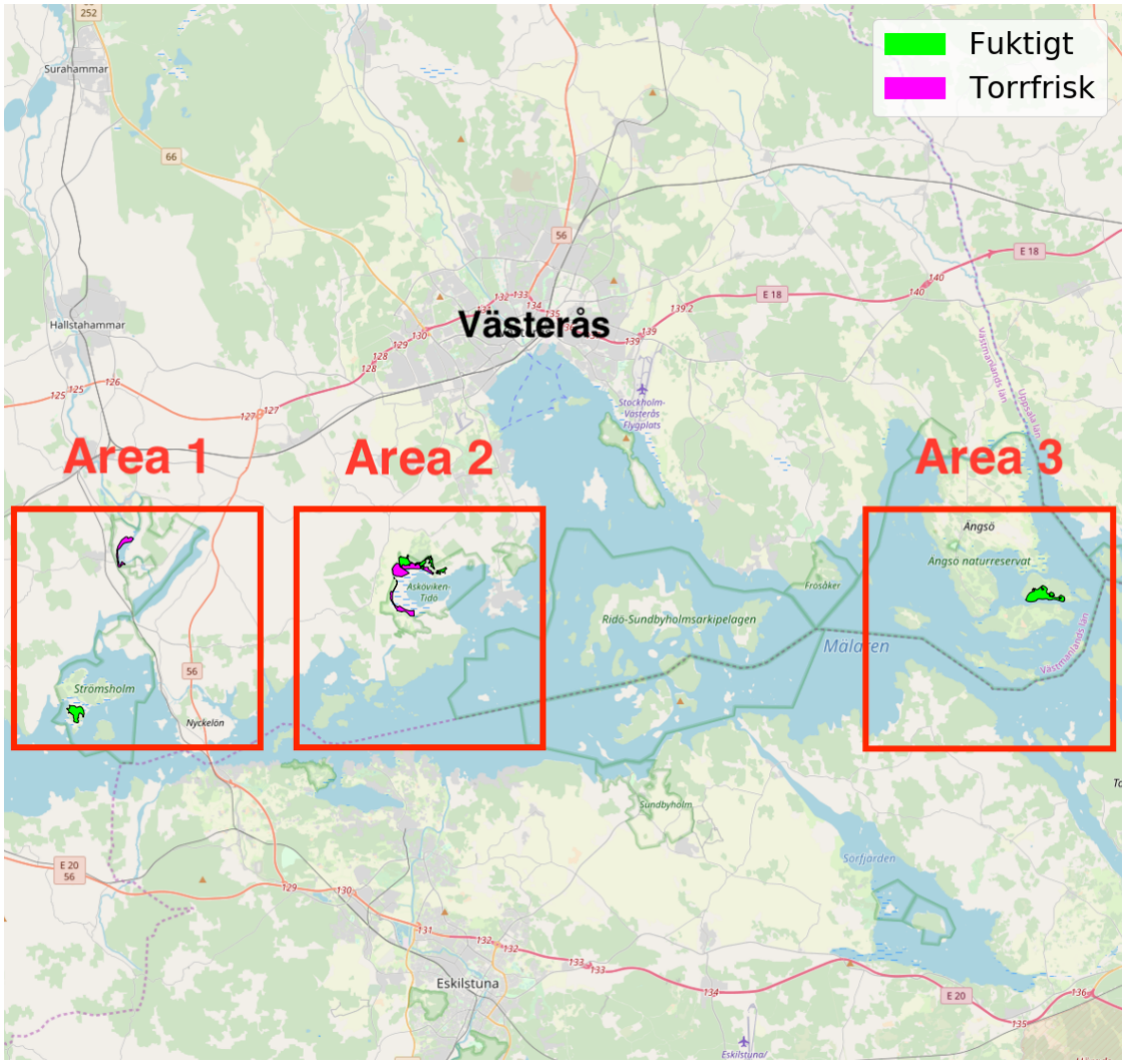
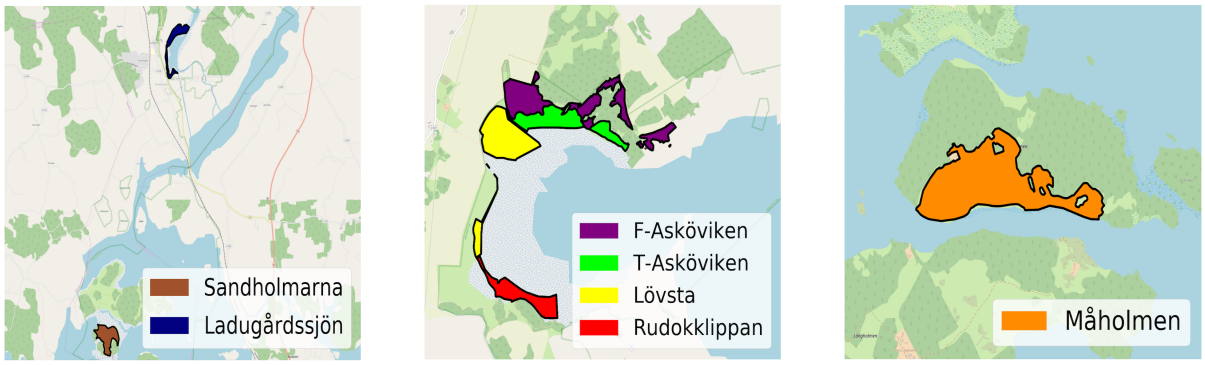



*Figure 3.2: Area of Interest and Loaded Area from the Datacube.*

Figure 3.3 (a) has two areas of interest named Sandholmarna and Ladugårdssjön plotted in brown and blue respectively. Figure 3.3 (b) has four areas, two of them named Asköviken but one classified as "fuktigt" (purple) and the other as "torrfrisk" (green). The other two areas' names are Lövsta and Rudökklippan which are plotted in yellow and red respectively. Figure 3.3 (c) has the last area of interest plotted in orange and named Måholmen.




(a) Area 1. (b) Area 2. (c) Area 3.

*Figure 3.3: Zoomed Areas.*

Table 3.2 summarizes the classification of the seven areas of interest.

| **Fuktigt** | **Torrfrisk** |
|---|---|
| Asköviken | Asköviken |
| Måholmen | Ladugårdssjön |
| Sandholmarna | Lövsta |
| | Rudökklippan |

*Table 3.2: Classification of AOI.*

### 3.3.3 Querying the Datacube and Loading the Data

The datacube has a lot of information, it has all the products summarized in Table 3.1 corresponding to a large area in middle Sweden from 2018 up to date. In order to query the datacube a series of parameters are needed. Querying the datacube means to select certain parameters to narrow down the data from the datacube that will be loaded into that notebook for analysis purposes. For example, a date range might be selected, also a coordinates reference system or a resolution in which the data can be loaded. Last but not least, some coordinates have to be given to specify a local area.

For this thesis' purpose, the coordinates reference system that was used was sweref99tm with a resolution of 20m. It is also possible to choose specific bands of the Sentinel-2 to load in order to speed up the loading time so only the required bands were selected. The selected time period was between May and September, both inclusive, for 2018 and 2019 so two datasets were loaded for each year. In order to maximize the number of days covered by the satellite, those datasets were the combination of two products, number three and four, the corresponding to sweref99tm crs with 20m resolution, one corresponding to Sentinel-2A and the other to 2B.

The coordinates loaded were defined by a shapefile defining any of the polygons of the areas of interest. Figure 3.4 shows the polygon in blue for the example of Måholmen. The red square represents the area that was loaded from the datacube in order to analyse the area in blue. To define the red square, the polygon shapefile was read, then its boundaries were identified together with the longest side. The rule was that the side of the square had to measure at least 2000m in order to achieve an acceptable resolution when plotting.


*Figure 3.4: Area of Interest and Loaded Area from the Datacube.*

The function in charge of finding the coordinates of the red square was named padding. It checked that the longest side of the polygon measured at least 2000m, if it was so, it padded a 5% of that length on each side and made the other side equal. In the case that the side was less than 2000m, the distance padded was the remaining until the side reached that length. The shortest side was padded to be equal and create a square. Figure 3.4 shows the result.

Table 3.3 summarizes the two loaded datasets that were named Dataset 1 and Dataset 2. Both contained data between May and October inclusive but the former was for 2018 and the latter for 2019. They were loaded combining two products of the Datacube, in other words, combining Sentinel-2A data with Sentinel-2B. Both used the Swedish coordinate reference system and 20m resolution.

| | Sentinel 2A | Sentinel 2B | May - Oct |
|---|---|---|---|
| **Dataset 1** | ✓ | ✓ | 2018 |
| **Dataset 2** | ✓ | ✓ | 2019 |

*Table 3.3: Creation of Dataset 1 and Dataset 2.*

### 3.3.4 Creating Masks and Weight Matrices

#### Masking the Area of Interest

The purpose of this step was to create masks or the so-called weight matrices that would help filtering between useful or invalid data. For example, the area of interest had a certain shape but the data loaded was a square around it, therefore a mask to identify that area was needed. Figure 3.5 shows the mask which consisted of a squared matrix for a certain time. That matrix in JupyterLab had the form of an xarray which had ones for the pixels that fell inside the area of interest and zeros for the rest. This way, a command could be used specifying to only take into account pixels where the mask equaled one.

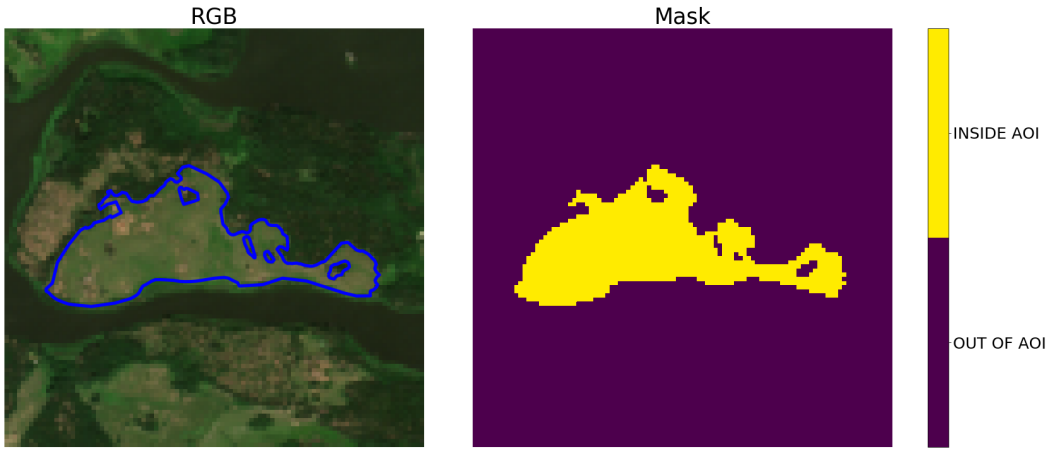



*Figure 3.5: Left: RGB Image with Polygon Shape. Right: Mask for the Area of Interest.*

#### Invalid Pixels Due to Satellite Sweep

Satellites orbit around the Earth and sweep its surface collecting data. The satellite's orbit is independent of the area being studied and it could be given the case that when the satellite swept, the selected area was at the border and did not get fully covered. That case would look like shown in Figure 3.6. A mask that identified those pixels was also created.

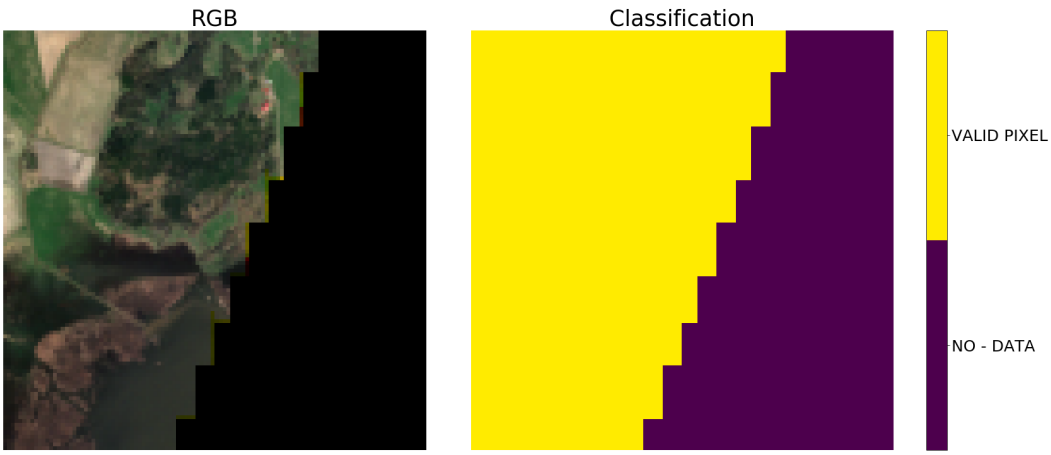



*Figure 3.6: Left: RGB Image. Right: Classification of Valid/Invalid Pixels due to Satellite Sweeping.*

### Scene Classification Layer

Satellites collect large amounts of data but not all of it might be useful for analysis purposes. The Sentinel-2 data at the Swedish Space Data Lab was downloaded from the European Space Agency. It is possible to download either Level-1C data or Level-2A. With the Level-2A product a Scene Classification Layer is provided. Figure 3.7 shows the convention values, colours, interpretations and labels.

| Label | Classification |
|---|---|
| 0 | NO_DATA |
| 1 | SATURATED_OR_DEFECTIVE |
| 2 | DARK_AREA_PIXELS |
| 3 | CLOUD_SHADOWS |
| 4 | VEGETATION |
| 5 | NOT_VEGETATED |
| 6 | WATER |
| 7 | UNCLASSIFIED |
| 8 | CLOUD_MEDIUM_PROBABILITY |
| 9 | CLOUD_HIGH_PROBABILITY |
| 10 | THIN_CIRRUS |
| 11 | SNOW |

*Figure 3.7: Scene Classification Values. [17]*

As seen in Figure 3.7 it is possible to identify pixels and classify them as clouds, snow, water, vegetation, bare soil, etc. The Scene Classification Layer also shorted as SCL band was one of the most important or useful bands for this analysis.

Figure 3.8 shows the use of the Scene Classification Layer (SCL) to identify pixels that were valid for the analysis and discard the ones that were not. The first picture in Figure 3.8 is a RGB image used to compare the actual aspect of the landscape with the classification obtained in the second picture. This case corresponds with the area of interest called Sandholmarna. It was a good candidate to show the use of the SCL band because it had clouds, vegetation and water properly identified. The legend for the SCL image is Figure 3.7 itself.

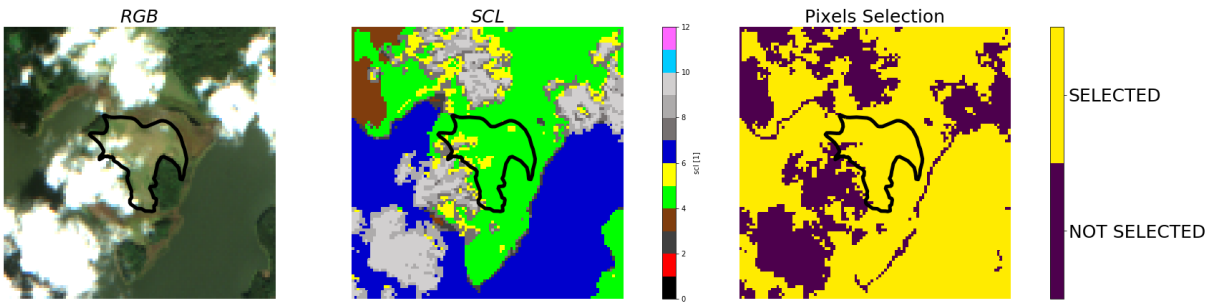



*Figure 3.8: Left: RGB Image. Middle: SCL Classification. Right: Pixels Selected as Valid.*

The right image from Figure 3.8 shows the mask created using the SCL band. The analysis consisted of the calculations of basically two indices and it was reasoned that it would not make sense to calculate the NDVI over a pixel classified as cloud or snow. Other pixel classifications such as no data, saturated or defective, dark area, cloud shadows or unclassified were also discarded for analysis purposes. Therefore, a mask that contained only pixels classified as water, vegetation or not vegetated (bare soil) was created. When comparing the second and third images of Figure 3.8, a correlation between selected pixels and red-green-blue pixels in the SCL classification was observed. The clouds or other invalid pixels were masked out.

### Weighted Matrices

Recapitulating, three different masks were created in order to identify the pixels that would not be used for the analysis. Those masks were also called matrices because when working with xarray all the masks information was presented in arrays or matrices that usually had more than two dimensions.

The first one of them was used to select only the area inside the polygon that represented the area of interest ($W_{AoI}$), the second one to identify pixels that had not been swept by the satellite ($W_{sweep}$) and the third one, obtained using the SCL band, was used

to identify pixels classified as vegetation, water or non-vegetation ($W_{scl}$). The combination of all three of them gave the total weight matrix, W:

$$W = W_{AoI} \cdot W_{sweep} \cdot W_{scl}$$

### 3.3.5 Cloud Free Dataset

The datasets loaded from the datacube contained all the dates in the period selected but some days were so cloudy that the satellite images were entirely covered by clouds. During those days the reflectance measured by the satellite was the one reflected from the clouds and not from the Earth surface. As a result, if those days were included in the analysis, false values or results would be obtained.

Figure 3.9 shows the NDVI time series during a certain period for two different datasets, one with clouds and one without. It is clearly appreciated how the cloud free dataset follows a trend, whereas the dataset with cloudy days has an unpredictable trend. This is the reason why a cloud free dataset had to be created.

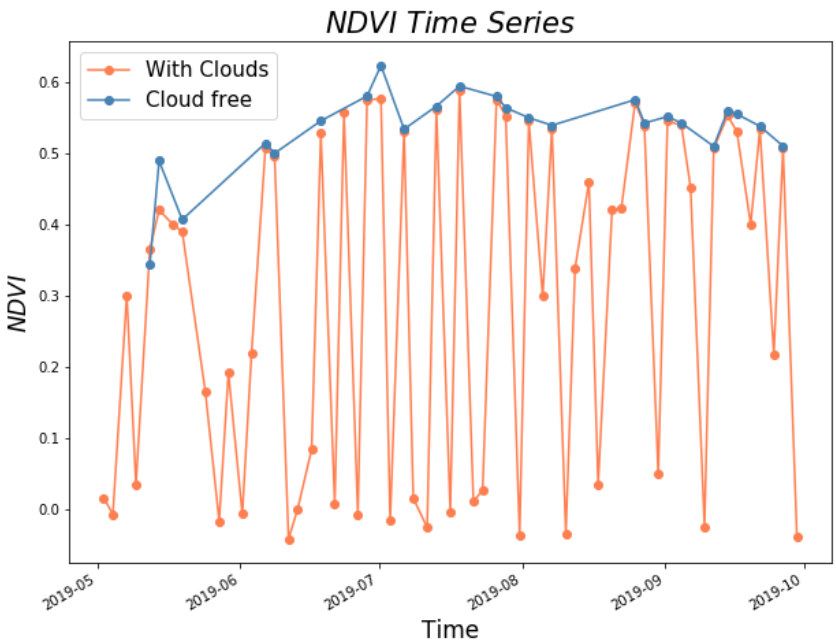



*Figure 3.9: NDVI Comparison Between a Cloud Free and a Non Cloud Free Dataset.*

A function named remove_clouds was created in order to identify cloudy days and remove them from the original dataset. It used the weighted matrices previously created

to check what percentage of cloud free or valid pixels were inside the area of interest. In the case that more than 80% of it was clear of clouds or invalid pixels, the date was added to a new dataset called filtered dataset (fds).

The resulting filtered datasets were double checked visually plotting all RGB images next to the SCL classifications. No cloudy days were observed for the remaining dates.

### 3.3.6 NDVI and MSI

NDVI and MSI were calculated using the masks $W_{scl}$ and $W_{sweep}$ so all the invalid or useless pixels were masked and the indices covered only valid pixels. Table 3.4 summarizes the link between the indices formulas, the Sentinel-2 associated bands, their central wavelengths and the SSDL name convention for those bands.

| | **Formula** | **S-2 Band** | **S-2 Central $\lambda$ (nm)** | **SSDL Name** |
|---|---|---|---|---|
| **NDVI** | $\frac{NIR - Red}{NIR + Red}$ | B8A (NIR)<br>B4 (Red) | 864.7<br>664.6 | nir_2<br>red |
| **MSI** | $\left[\frac{1600\ nm}{820\ nm}\right]$ | B11<br>B8A | 1613.7<br>864.7 | swir_2<br>nir_2 |

*Table 3.4: NDVI and MSI Bands Selection Summary.*

The Near Infrared (NIR) band had two possible choices of bands: band 8 or 8A. The main difference between both of them was the bandwidth, being B8A narrower by 85nm. They both were centered at the near infrared at 832.8nm for B8 and 864.7 for B8A. The reason why B8A was chosen was because most studies used B8A due to it being more comparable to the Landsat-8 band (B5) corresponding to NIR. [18] [19] [20]

Figure 3.10 shows an example of NDVI and MSI plots. The scale for MSI was reversed so that both had green associated to moisturized areas. There is a clear correlation between MSI and NDVI, the same areas that have low NDVI present high MSI which means water stressed areas.

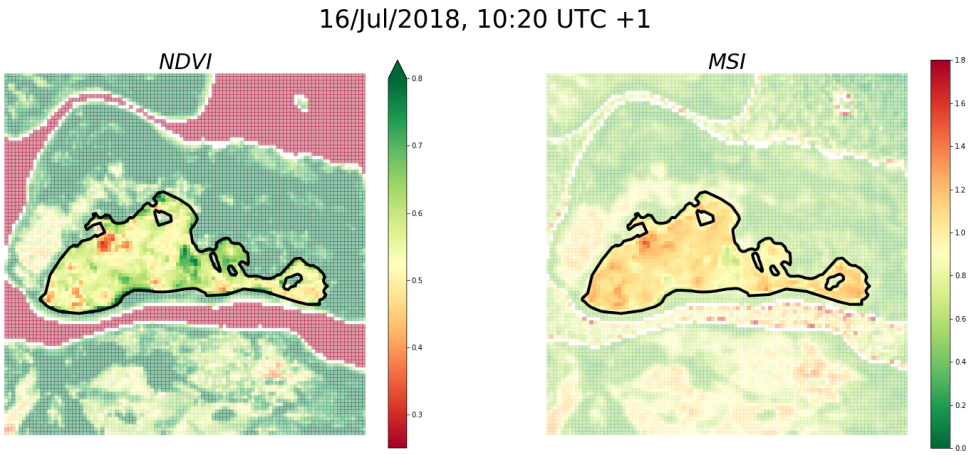



*Figure 3.10: NDVI and MSI Examples.*

## 3.4 Jupyter Notebooks Analysis

The analysis purpose for the Pilot Mälardalen was to display three different views, being the first one a comparison between the same period over two different years. The second view was a time trend for both years using machine learning techniques and the third one consisted of an interactive display. It had a play button and a slide bar made using ipywidgedts which offered the possibility of selecting any day within the dataset to be displayed.

### 3.4.1 Gaussian Process Regression

The technique of Gaussian Process Regression was useful to average the indices NDVI/MSI over time, filling the gaps between observations. The goal was to present the trends in vegetation for seven areas of interest so that they could be followed through a season or compared with other years.

The four steps in order to get a GPR were: dataset preparation, model construction or training, prediction and finally plot. The online documentation on GPflow was really helpful to understand the implementation in Python. [21]

#### Dataset Preparation

As described in Section 2.5, the dataset needed to be composed by the matrix $X$ and the target vector $Y$:

$$X = \begin{bmatrix} t_0 & y_0 & x_0 \\ t_1 & y_1 & x_1 \\ \vdots & \vdots & \vdots \\ t_n & y_n & x_n \end{bmatrix} \quad , \qquad Y = \begin{bmatrix} I_0 \\ I_1 \\ \vdots \\ I_n \end{bmatrix} \tag{3.1}$$

The matrix $X$ contained the input data which were the coordinates *time,* $y$ and $x$ arranged in three vectors. The vector $Y$ had the output data, in this case the NDVI/MSI values corresponding to the coordinates in $X$.

The initial idea was to perform a Gaussian Process Regression for each of the seven areas of interest separately and train the hyperparameters. However, it was observed that doing the training for the seven areas together gave more stable results.

Using the pre-processing notebook described in Section 3.3, fourteen cloud-free datasets together with their weight matrices were pickled and loaded at the beginning of this notebook. Fourteen cloud-free datasets means one dataset per year and area, that is seven areas and two years, 2018 and 2019.

The first function was called *load_pickles* and it had two inputs, the area name and the index to be calculated, NDVI or MSI. Its function was to load the two cloud-free datasets corresponding to that area and their weight matrices, then combine the matrices into one, $W = W_{scl} \cdot W_{sweep} \cdot W_{AoI}$. At last, it would call another function named *calculate_index* which decided whether the function *NDVI* or *MSI* should be called. Those functions calculated the indices over valid pixels and inside the area of interest. The function *load_pickles* returned two xarrays, one per year, with the corresponding values of the selected index.

It is possible to model Gaussian processes in more than one dimension. In this case, there were three dimensions, two spatial coordinates creating images 2D over time. Therefore, the possibilities were to either calculate time trends predicting averaged values for an area only in time, or to predict in time and space, for each pixel of the image and then use the values to plot time trends or maps. For the second choice the dataset size would increase and Gaussian Processes might not perform so well. More advanced methods such as Sparse Gaussian Process Regression with the use of Variational Bayes were explored, but decided to be out-of-scope for the purpose of the master thesis as good results were obtained also with Gaussian process that predicted only the time-series.

After obtaining the xarray that contained the index values in the three dimensions it was necessary to average them over $x$ and $y$. The function named *conc_xy* did so, keeping one x-coordinate and one y-coordinate which means that every image got a mean value with a location in the middle of the area of interest. This was needed in order to train the dataset for the seven areas together.

Finally the function *collect* created a DataFrame with the columns: area name with year, time in date-time format, y-coordinates, x-coordinates, index values and time in

time-stamp format. Subtracting the columns time-stamp, y and x, *X_train* was defined. The same way, subtracting the column for the index, *Y_train* was created.

### Model Construction

In this step, a model for the seven areas together was trained. Training a model means finding the optimal values of the hyperparameters and that is already implemented in Python. For this thesis, it was only necessary to define $X_{train}$, $Y_{train}$, a kernel and a mean function (See Section 2.5.3). The values of the hyperparameters could be set to some pre-defined values if needed but it was not the case.

The mean function was set to constant with an initial value of 0.5. That means its initial value was set but at the end of the training Python had calculated a different value. It kept being constant but with another value. The chosen kernel was *RBF* or also known as Squared Exponential. It was defined in three dimensions with the ARD option set to True allowing to have a length-scale parameter per dimension.

### Prediction

The last step before plotting was to define some points distributed between the data points. Those points were called prediction points. Then, the value of the mean function with its variance had to be calculated for the prediction points. An offset of 30 days before and after the first and last data points was set. A total of 100 prediction points were distributed evenly in the defined period. This process was done inside a function called prediction function.

### Plot

The mean value and the variance were the outputs of the prediction function. Those outputs were needed to plot the GPR. The mean value was plotted as a continuous line and the variance was used to plot confidence bands. The errors obtained for the distribution represented the uncertainty in the measurements and it could be visualized in the confidence bands. Two times the standard deviation equals to a confidence band of 95%. The original data points were plotted as crosses.

# Chapter 4

# Results & Discussion

## 4.1 Pilot Mälardalen

The analysis purpose for the Pilot Mälardalen was to display three different views, being the first one a comparison between the same period over two different years. The second view was a time trend for both years using machine learning techniques and the third one was a day-by-day displayed with a play button and a slide bar using ipywidgedts.

Figure 4.1 is an example of the first view. It had three subplots, the first and second were the averaged RGB bands for 2018 and 2019 respectively. The average was taken over all the available dates for the selected period. Therefore, the first and second images were presented as representative plots for the user to be able to get a visual feeling of greenness during those periods.

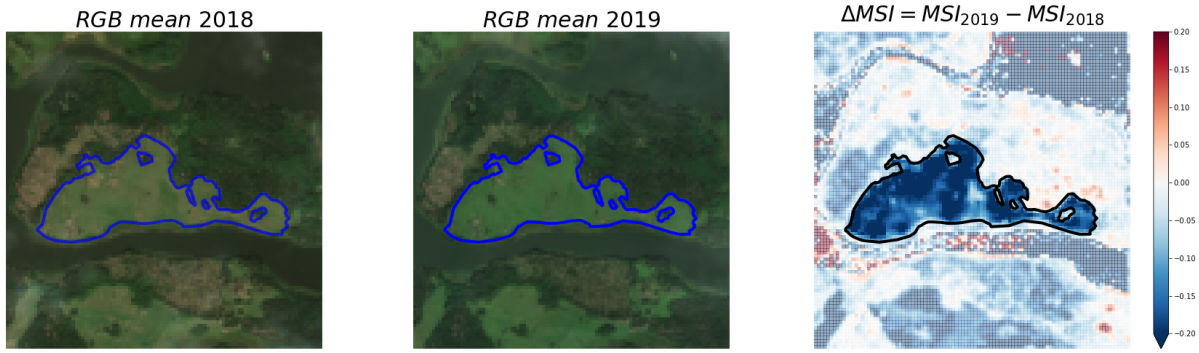



*Figure 4.1: Pilot Mälardalen Example of a First Analysis View.*

The NDVI and MSI indices were calculated for both years and averaged. In order to compare them the difference was plotted in the third subplot. The chosen scale was such

that identified drier areas with red and wetter areas with blue.

Figure 4.1 is an example view of the area named Fuktigt Måholmen with MSI analysis. It can be appreciated that the image for 2019 looks greener than the one for 2018 which indicates that 2018 was a drier year. In the third plot, the difference of MSI between both years appears blue which in the case of MSI, means a negative difference. Since high values of MSI translate to high water stress levels, the year 2018 must have had higher water stress values than the year 2019 in Måholmen. Hence, the second year, 2019, was wetter and that is represented by blue. In the case of NDVI the difference would be positive because healthier or hydrated vegetation have higher values of the index.

Figure 4.2 shows the time trends. The first and second subplot correspond to the time trends of 2018 and 2019 respectively. The third subplot is both years combined. A Gaussian Process Regression was used in order to predict the trends between the data points. This particular view is produced using methodology improved upon since the end of November when the report for the Mälardalen project was presented. After that date, the thesis work focused on Gaussian processes time-series regression and better results could be obtained compared to the project report.

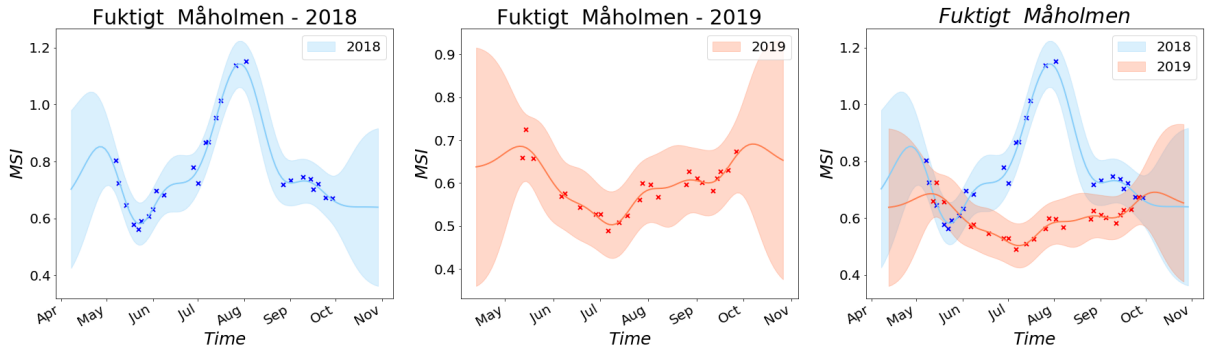



*Figure 4.2: Pilot Mälardalen Example of a Second Analysis View.*

The three subplots shown in Figure 4.2 also correspond to the area named Fuktigt Måholmen to continue with the same example. The results of MSI time trends support the observations from the first view. Higher water stress is represented by high MSI values and in Figure 4.2 is clearly shown how 2018 had higher values from July until September. The difference is clear because the confidence bands separate themselves. In the case of overlap it would not be possible to determine any difference with confidence because the values could be within the same range.

The third view is shown in Figure 4.3. However, since it is a display widget, not all the features can be appreciated in this report. The idea for the last view was to present a RGB picture next to its corresponding NDVI/MSI image for every available date in the period. The RGB picture helped the user to interpret the index results when comparing

both images. The widgets allowed to display them date by date and offer the possibility of selecting a specific day or pausing the display. The scales for NDVI and MSI were chosen in a way that red was associated with dry, or water in the case of NDVI, and green meant wet or healthy vegetation.

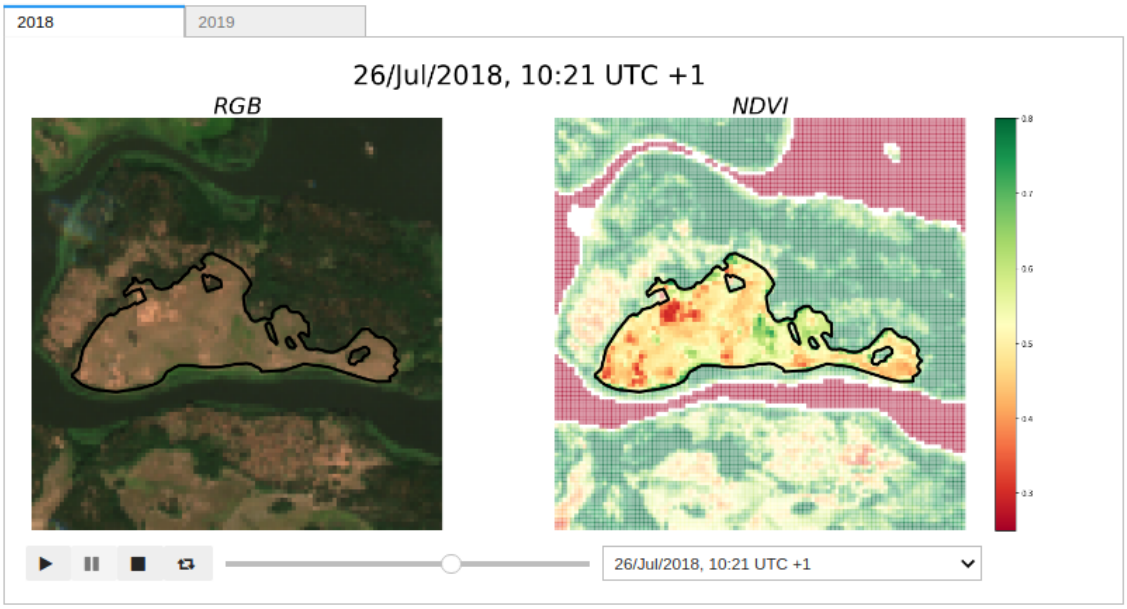



*Figure 4.3: Pilot Mälardalen Example of a Third Analysis View.*

The images displayed in Figure 4.3 also correspond to Fuktigt Måholmen. In this case the selected index was NDVI to show that the analysis worked independently of the index selection. The widget had two tabs, one per year, and both tabs had the same layout. The datasets had 26 and 24 days for 2018 and 2019 respectively and all of them were displayed in this view. The chosen date that appears in this report is 26th July 2018 which would be located at the MSI peak of the second view, see Figure 4.2. Other dates with lower values of MSI or higher NDVI looked greener.

There were two type of red pixels in the NDVI image, some corresponded to water and others to bare soil. Måholmen is an island and the water surrounding it might not be noticeable in the RGB picture. Nevertheless, water was represented as red in NDVI, having negative values. When the two images were compared, pixels corresponding to soil appeared in light brown for the RGB image whereas water was dark green.

Comparing the two subplots of Figure 4.3 the correlation between NDVI values and visual aspect of the ground could be seen. Lower values of NDVI represented dry soil or unhealthy vegetation which were shown as red pixels. Those red pixels had actually a different color in the RGB image than the rest of the soil. It could also be deduced that there was no vegetation in those pixels, as generally, the greener the pixels appeared in

the RGB image, the darker the green of the NDVI value.

## 4.2 Gaussian Process Regression

The Gaussian Process Regression was done to predict time trends in all seven areas of interest together with confidence bands. The model was trained for the seven areas combined because it gave more stable results. The parameter values after training the model were summarized in Table 4.1:

| **Parameters** | **Trainable** | **Value** |
|---|---|---|
| Time length-scale ($\ell_t$) | True | 14.879 days |
| y length-scale ($\ell_y$) | True | 0.1982 km |
| x length-scale ($\ell_x$) | True | 1.4536 km |
| Kernel variance | True | 0.0065 |
| Likelihood variance | True | 0.0009 |
| Mean function | True | 0.7368 |

*Table 4.1: Parameter Values of the GPR Model.*

All the parameters were set to be trained which means that Python was in charge of finding their optimal values. The mean function reached a reasonable mean value for all the time trends. The likelihood variance corresponds to the noise variance and its value was quite small. The found value for kernel variance or signal variance was also low.

The most interesting parameters for this study were the length-scales, in particular the time length-scale. Its value was found to be 14.879, which meant that a particular data-point had a correlation with other data-points to be predicted 15 days before or after it. That translated to a smoothing of the time trend and less wiggly functions. When the model was trained separately for each area there was no consistency on the time scales and the predicted trends looked very different.

The spatial length-scales played a less important role than the time one. A central reference coordinate was introduced for $x$ and $y$ in order to be able to make a 3D GP regression but the goal was not to study spatial correlation. Table 4.2 summarizes the selected coordinates in km.

The distance between areas in the x-direction was of the order of 20 km and the x length-scale value was 1.45 km which meant that there would not be any influence between areas. In the case of y-direction the distance between areas was of the order of 1km and the length-scale parameter 0.19, an order of magnitude smaller. The time length-scale was 15 days and the time between data-points could be 3-4 days, therefore there was an important time correlation.

| Area | X (km) | Y (km) |
|---|---|---|
| F-Asköviken | 583.3 | 6599.53 |
| F-Måholmen | 608.79 | 6598.99 |
| F-Sandholmarna | 569.19 | 6593.15 |
| T-Asköviken | 583.3 | 6599.36 |
| T-Ladugårdssjön | 571.08 | 6599.81 |
| T-Lövsta | 582.37 | 6598.79 |
| T-Rudökklippan | 582.51 | 6597.74 |

*Table 4.2: Reference Coordinates in SWEREF of the Areas Used for GPR.*

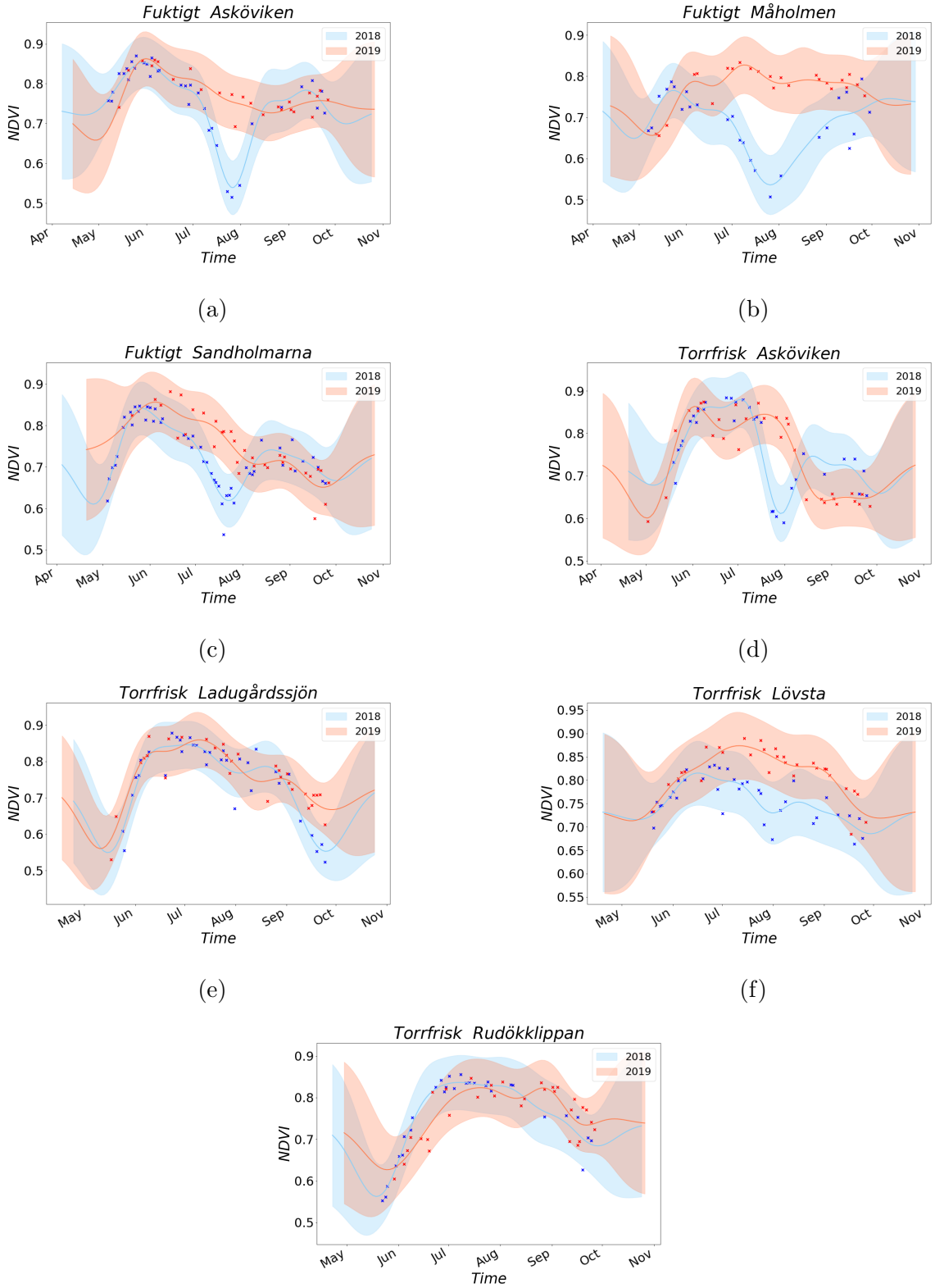
Figure 4.4 shows the results of the NDVI time trends obtained for 2018 and 2019 in all seven areas. All 2018 data-points were represented by blue crosses whereas all 2019 by red crosses. The shadowed areas represented the confidence bands and their color was associated with their corresponding year color. The blue and red continuous lines represented the mean predicted values of the GPR.

All trends started with lower values of NDVI around April-May probably because the vegetation had been covered by snow during winter and the trees might have lost their leaves. Then the levels rose up for summer when everything looked green. After September the normal trend was to decrease again. Most of the areas followed that trend for 2019.

The year 2018 was a bit exceptional because of the heat waves recorded all over Europe. Some of these areas presented a dip in NDVI around the end of July/beginning of August. It was noticeable that all the dips occurred around the same days in 2018. When the confidence bands between two years clearly separated, it was possible to say that one year was drier than another because even within the margin error, the index values would not be similar. That is the case of Fuktigt Måholmen, Fuktigt Asköviken or Torrfrisk Asköviken.

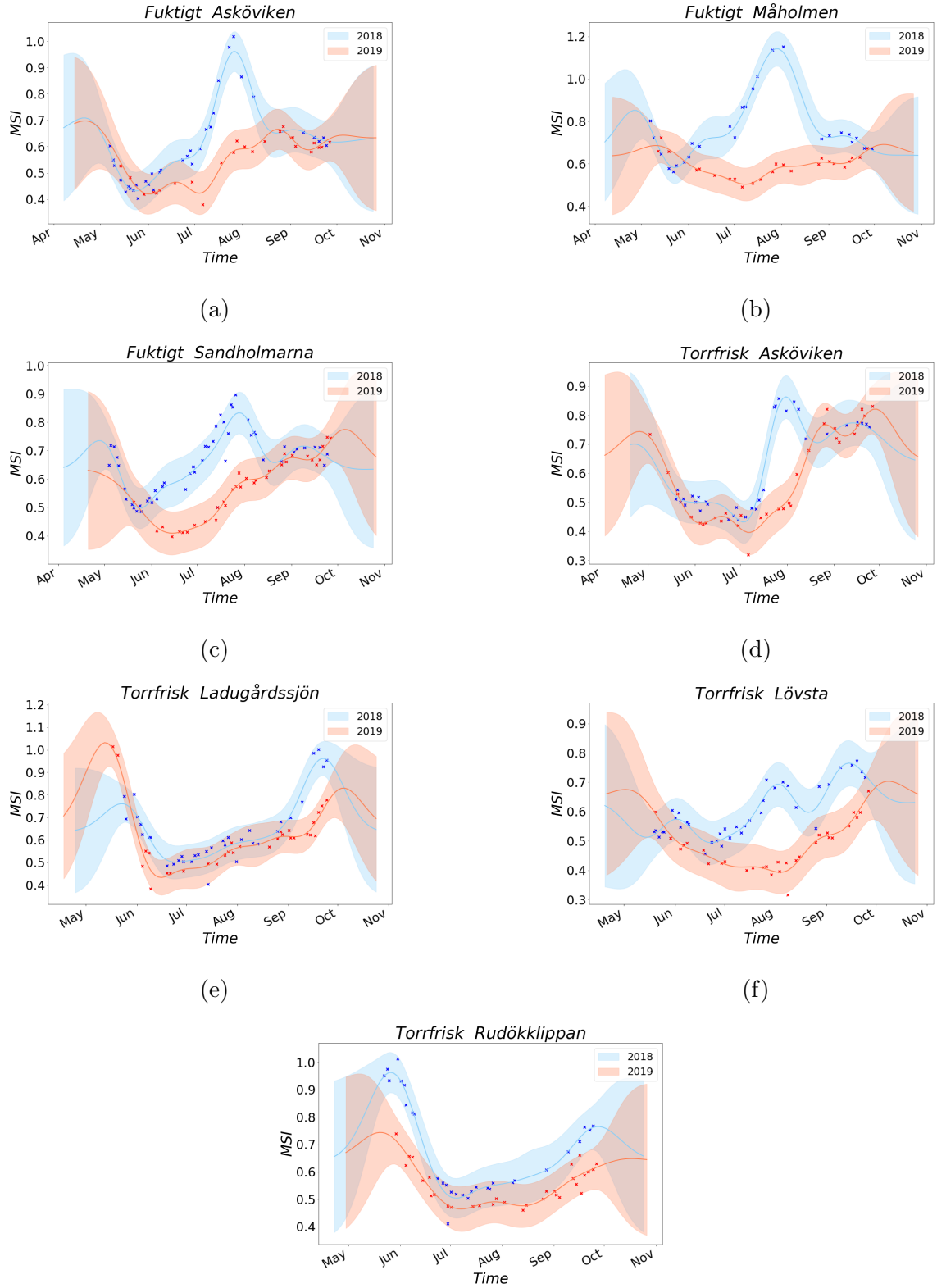
Figure 4.5 shows the same plots as Figure 4.4 but for MSI instead of NDVI. Since high MSI values equal to low values of NDVI, where the NDVI graphs had dips, the MSI presented peaks. That was the case of the areas Fuktigt Måholmen and both Asköviken. The confidence bands clearly separated in those cases and the dates corresponded also to the end of July or beginning of August.

Other areas like Sandholmarna of Lövsta had slightly lower values of NDVI for 2018 than for 2019 but in the case of MSI those differences could be appreciated easier with clearly separated confidence bands. In general, the year 2018 was drier than 2019.




(a) (b) (c) (d) (e) (f) (g)

*Figure 4.4: NDVI Trends in the Seven Areas of Interest.*




(a) (b) (c) (d) (e) (f) (g)

*Figure 4.5: MSI Trends in the Seven Areas of Interest.*

## 4.3 Validation of the Method

In order to validate the method used for the GPR a method called cross-validation was used. The idea of this method was to split the dataset into two datasets, training and validation. From the 450 total data points, 50 were randomly removed and became the validation dataset. The remaining 400 were used to train the model. Then the mean value of NDVI/MSI at all 450 points was predicted. After that, the predicted values could be compared with the real values in the dataset calculating the errors RMSE and MAE. If the point was in the training dataset the error was called in-sample and if it was in the validation, out-of-sample error.

That process was repeated for 1096 times in order to get statistical values of the distribution created by the parameters of the different trained models. This provided information about the robustness of the training procedure. Table 4.3 summarizes the distributions of the three length-scales, the covariance or kernel variance and the likelihood variance. It tells what are the mean, minimum and maximum values, the standard deviations and the percentiles 25,50 and 75% which equals to a numerical description of the histograms.

| | $\boldsymbol{\ell_t}$ (days) | $\boldsymbol{\ell_y}$ (km) | $\boldsymbol{\ell_x}$ (km) | **cov** | **lkv** |
|---|---|---|---|---|---|
| **count** | 1096 | 1096 | 1096 | 1096 | 1096 |
| **mean** | 14.9309 | 0.1871 | 1.6939 | 0.0064 | 0.0009 |
| **std** | 0.5905 | 0.0348 | 0.7633 | 0.0002 | 0.0000 |
| **min** | 9.4405 | 0.0072 | 0.6346 | 0.0031 | 0.0007 |
| **25%** | 14.5575 | 0.1845 | 1.1686 | 0.0063 | 0.0009 |
| **50%** | 14.8968 | 0.1912 | 1.3649 | 0.0065 | 0.0009 |
| **75%** | 15.2649 | 0.1992 | 1.8595 | 0.0066 | 0.0009 |
| **max** | 17.2602 | 0.3697 | 3.2137 | 0.0071 | 0.0015 |

*Table 4.3: Statistics of the Model for 1096 Cases.*

The mean values of NDVI/MSI were close to the values obtained for the model using all 450 points described in Table 4.1. The values of the percentiles are better seen in box plots like shown in Figure 4.6. The 50th percentile equals the median of the distribution and it was represented by a golden line in the middle of the boxes. The 25th and 75th percentiles correspond to the edges of the boxes. The lines going out from the boxes are called whiskers and represent a 24.65% of the distribution on each side, leaving only a 0.35% out.

Figure 4.6 was plotted with two y-axes. The one on the left was used for the three length-scales and the color used was blue. The axis on the right used green and it was used for the covariance and the likelihood variance. As mentioned before, the variance of the spatial length-scales was not important compared to the distance between areas.

The time length-scale had a standard deviation of 0.6 days which is not so big given that the time span between data points could be 4-5 days.

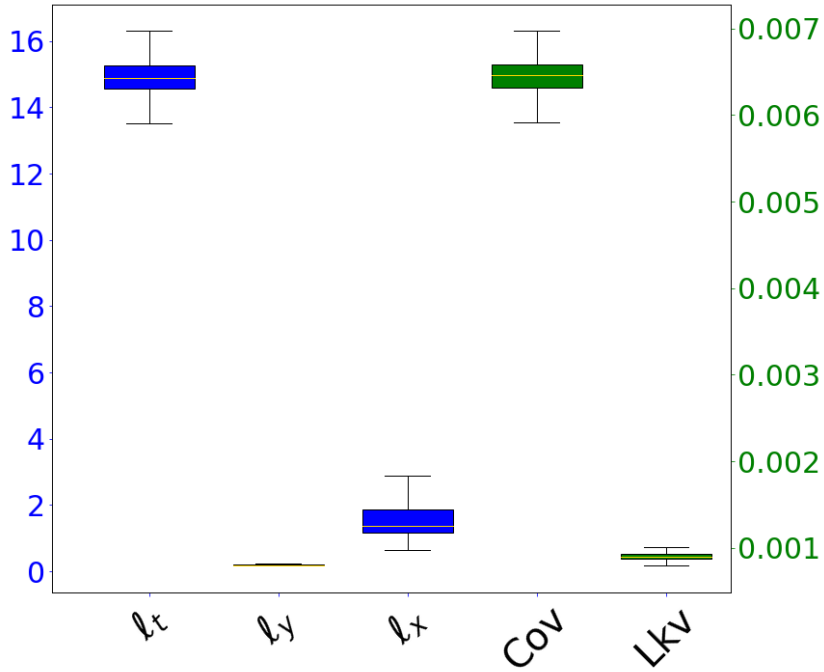
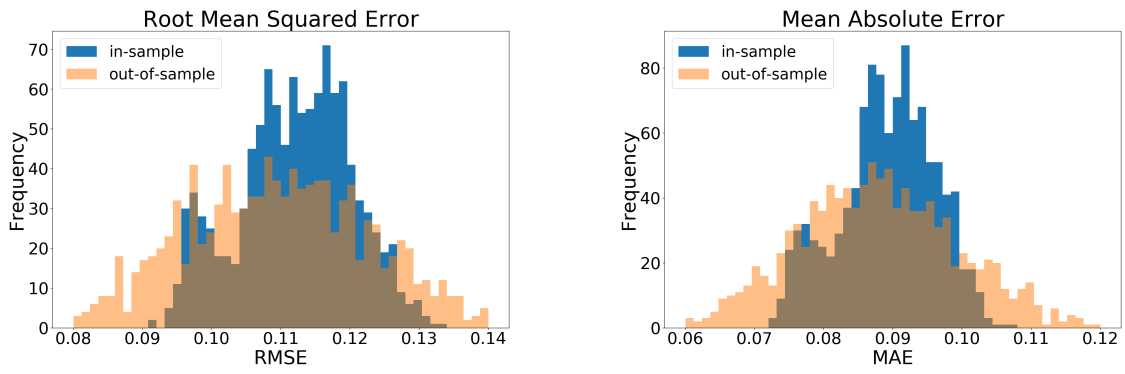



*Figure 4.6: Box Plot of the Distributions of the Hyperparameters.*

After training 1096 models, the predictions trends were made for all 450 data points. Those values were compared to the real values in the dataset calculating the Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE). The Mean Absolute Error was used in order to minimize the effect of the outliers. A differentiation was made between in-sample error and out-of-sample error, meaning that the former belonged to the training dataset points and the latter to the validation dataset points.




(a) (b)

*Figure 4.7: RMSE and MAE Histograms for In-Sample and Out-of-Sample Errors.*

Figure 4.7 shows the histograms for both, RMSE and MAE, distinguishing between in and out-of-sample errors. The distribution for in-sample errors was sharper than out-of-sample because the models were trained with those data points.

The optimization problem to determine kernel parameters always tries to find the best values. If any kernel was possible, the curve would fit perfectly all the data points and the in-sample error would be zero. This is not the case but when optimizing, the kernel values obtained are those which minimize the in-sample error.

Since the models were trained with particular data points, it was expected that the in-sample error for those points would be smaller than the out-of-sample. However, that was not what the results showed when comparing the average values of RMSE and MAE for all the models, the numbers were very close and there was no significant difference. Table 4.4 summarizes the numbers.

| | **RMSE** | | **MAE** | |
|---|---|---|---|---|
| | **Mean** | **Std** | **Mean** | **Std** |
| in-sample | 0.112 | 0.008 | 0.089 | 0.007 |
| out-of-sample | 0.110 | 0.013 | 0.088 | 0.011 |

*Table 4.4: Mean and Standard Deviation of MAE and RMSE.*

Given that the out-of-sample error did not turn out to be much larger than the in-sample, the method used was validated showing that there was no over-fitting.

# Chapter 5

# Conclusions & Future Work

## 5.1 Conclusions

This thesis contributed to increase the use of space data. It was a part of the start up of the Space Data Lab in Sweden and the analysis tools developed during the thesis work are now part of it. The main contributions of this thesis to the Space Data Lab were a set of Jupyter Notebooks that will allow future users to get introduced with the DataCube environment and its data, load datasets and filter out cloudy days, calculate indices like NDVI or MSI and apply basic machine learning algorithms such as Gaussian Process Regressions to visualize time trends.

Those tools were the first services the Space Data Lab was able to provide and they are now available to governments or Swedish authorities, ready to use for different purposes such as environmental, sustainability, development of society, etc.

The results obtained for the Pilot Mälardalen project showed time-series with confidence bands that separated between the two years for most of the analyzed areas. The methodology chosen was appropriate to compare year on year with respect to indicators of drought and showed that 2018 was generally drier than 2019 . However, a comparison between two years is not enough to draw conclusions on climate change and more data is needed to study trends.

## 5.2 Future Work

During this thesis work some interesting topics that could need more research were encountered. Since they were beyond the scope of this project, they could be considered future work or continuation of this thesis.

The current ESA algorithm used to provide the Scene Classification band (SCL) that classifies pixels is not perfect and could need some improvements. It is an algorithm de-

signed for Europe and when focusing on Sweden which is at the north, some assumptions might be too general and not useful for Sweden in particular. There were some problems sometimes with cloud identification and more commonly thin cirrus. It was also difficult sometimes for the algorithm to differentiate between snow and clouds. Besides that, it was accurate most of the time and this analysis relied on it most of the time. Future work to improve this could be done using for example deep learning.

The Gaussian Process Regression was done to predict mean values of an index in time but the next step would be to not only predict in time, predict in space too. The idea would be to predict entire images between two data points or partially reconstruct images were there are clouds or invalid pixels. GPR is a trusted method for small sample size problems but has well known limitations for large data sets such as images since time complexity scales as $N^3$. One solution for this that could be further explored is using Variational Bayes to find a sparse GPR that approximates to the original problem. Alternatively, solutions based on deep learning with convolutional neural networks could be explored as such has demonstrated good performance on similar problems, while scaling better than GPR with the size of the problem

One of the main issues had during this thesis work was that the Swedish Space Data Lab project started at the same time as the pilot Mälardalen. That implied that the development of the platform had to be done at the same time the first users started using it. As the platform was on early stages of development, it was not stable enough at all times and the experience of this master thesis was used as feedback to continue developing the platform. The SDL needs further work to improve its stability and more data shall be added.

# References


[1] Open Data Cube. https://medium.com/opendatacube.

[2] Australian Government - Geoscience Australia, Committee on Earth Observation Satellites, Commonwealth Scientific and Industrial Research Organisation, United States Geological Survey, Catapult Satellite Applications, and Analytical Mechanics Associates. Open Data Cube. https://www.opendatacube.org/.

[3] A. Lewis, S. Oliver, L. Lymburner, et al. The australian geoscience data cube — foundations and lessons learned. *Remote Sensing of Environment*, 202:276 – 292, 2017. Big Remotely Sensed Data: tools, applications and experiences.

[4] ESA. Copernicus Overview. https://www.esa.int/Our_Activities/Observing_the_Earth/Copernicus/Overview3.

[5] European Comission. Discover our satellites. https://www.copernicus.eu/en/about-copernicus/infrastructure/discover-our-satellites.

[6] Missions - Sentinel Online. https://sentinel.esa.int/web/sentinel/missions/.

[7] A Landsat Timeline - Landsat Science, howpublished=https://landsat.gsfc.nasa.gov/a-landsat-timeline/, journal=NASA, publisher=NASA.

[8] Swedish Space Data Lab - RISE. https://www.ri.se/en/what-we-do/projects/swedish-space-data-lab.

[9] NDVI and NDMI vegetation indices: instructions for use. https://www.agricolus.com/en/indici-vegetazione-ndvi-ndmi-istruzioni-luso/, Sep 2019.

[10] Chun Qi, Joy Ng, Yun Toh, Chee Yong, Leslie Lam, Cw Chang, and Soo Chin Liew. Effects of leaf water content on reflectance. *28th Asian Conference on Remote Sensing 2007, ACRS 2007*, 1, 01 2007.

[11] E. Ibarrola-Ulzurrun, J. Marcello, C. Gonzalo-Martín, and J. L. Martín-Esquivel. Temporal Dynamic Analysis of a Mountain Ecosystem Based on Multi-Source and Multi-Scale Remote Sensing Data. *Ecosphere*, 10(6), 2019.

[12] Sentinel Hub. https://www.sentinel-hub.com/develop/documentation/eo_products/Sentinel2EOproducts.

[13] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[14] H. Hultin. Evaluation of Massively Scalable Gaussian Processes. Master's thesis, KTH, Mathematical Statistics, 2017.

[15] A. G. de G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagra, Z. Ghahramani, and J. Hensman. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6, apr 2017.

[16] C. R. Rao and V. Govindaraju. *Handbook of Statistics, Volume 31: Machine Learning Theory and Applications*. North Holland & IFIP, 1st edition, 2013.

[17] ESA - Sentinel Online. Technical Guides. `https://earth.esa.int/web/sentinel/technical-guides/sentinel-2-msi/level-2a/algorithm`.

[18] S. Li, S. Ganguly, J. L. Dungan, W. Wang, and R. R. Nemani. Sentinel-2 MSI Radiometric Characterization and Cross-Calibration with Landsat-8 OLI. *Advances in Remote Sensing*, 06(02):147–159, 2017.

[19] T. Zhang, J. Su, C. Liu, W. Chen, H. Liu, and G. Liu. Band selection in sentinel-2 satellite for agriculture applications. In *2017 23rd International Conference on Automation and Computing (ICAC)*, pages 1–6, Sep. 2017.

[20] K. Z. Hankui, D. P. Roy, L. Yan, Z. Li, H. Huang, E. Vermote, S. Skakun, and J. C. Roger. Characterization of Sentinel-2A and Landsat-8 top of atmosphere, surface, and nadir BRDF adjusted reflectance and NDVI differences. *Remote Sensing of Environment*, 215:482 – 494, 2018.

[21] GP Regression with GPflow - gpflow 1.0.0 documentation. `https://gpflow.readthedocs.io/en/stable/notebooks/regression.html`.

[22] S. Kopp, P. Becker, A. Doshi, D. J. Wright, K. Zhang, and H. Xu. Achieving the Full Vision of Earth Observation Data Cubes. *Data*, 4(3):94, Jul 2019.

[23] L. Frau, S. Rizvi, B. Chatenoux, C. Poussin, J. P. Richard, and G. Giuliani. Snow Observations from Space: An Approach to Map Snow Cover from Three Decades of Landsat Imagery Across Switzerland. pages 8663–8666, 07 2018.

[24] B. Chatenoux, J.P. Richard, C. Poussin, Y. Guigoz, and G. Giuliani. Bringing Open Data Cube into Practice - Workshop Material, 01 2019.

[25] G. Giuliani, B. Chatenoux, A. De Bono, D. Rodila, J. P. Richard, K. Allenbach, H. Dao, and P. Peduzzi. Building an Earth Observations Data Cube: lessons learned from the Swiss Data Cube (SDC) on generating Analysis Ready Data (ARD). *Big Earth Data*, 1(1-2):100–117, 2017.