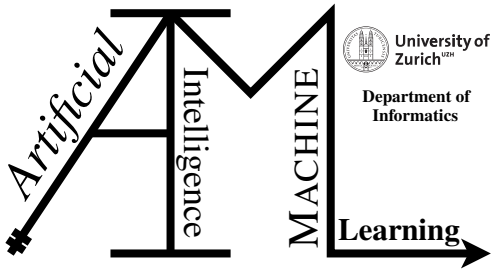
# Forest Drought Prediction based on Spatio-temporal Satellite Imagery and Weather Forecasts

Master Thesis

**Emine Didem Durukan**
20-745-733

**Submitted on**
February 22 2023

**Thesis Supervisor**
Prof. Dr. Manuel Günther
Dr. Thomas Brunschwiler, IBM Research Europe
Prof. Benjamin Stocker, University of Bern



**Master Thesis**

**Author:** Emine Didem Durukan, eminedidem.durukan@uzh.ch

**Project period:** 08/22/2022 - 02/22/2023

Artificial Intelligence and Machine Learning Group
Department of Informatics, University of Zurich

# Acknowledgements

I would like to thank Selene Ledain from École Polytechnique Fédérale de Lausanne for her invaluable contribution to this work. Moreover, I wish to thank Prof. Manuel Guenther and Prof. Benjamin Stocker from University of Bern, Dr. Thomas Brunschwiler from IBM Research Europe, and Prof. Devis Tuia from École Polytechnique Fédérale de Lausanne for their valuable insights and constructive critiques of my thesis. I would also like to thank IBM Research Cognitive Computing Cluster service for providing resources that have contributed to the research results reported within this thesis.

# Abstract

Considering the condition of our planet, anticipating natural disasters has long been a hot topic. This work is becoming more doable thanks to the expansion of earth observation data sources, such as satellite imagery. In this work, our main interest is droughts and their impacts. Recent hot and dry summers in Europe have had a significant impact on forest functioning and structure. In 2018 and 2019, Central Europe experienced two extremely dry and hot summers. These extremes resulted in widespread canopy defoliation and tree mortality. The objective of this study is to create a predictive model for forecasting future satellite imagery that contains information about the greenness of vegetation as measured by the Normalized Difference Vegetation Index (NDVI). We predict NDVI utilising data from the previous months as input to determine where and when drought impacts are triggered. We use a combination of temporal bands from Sentinel 2 and ERA-5 data sources, as well as static data sources such as the NASA SRTM Digital Elevation Model and the Copernicus Landcover Classification Map, as predictors. We will now focus on the forests of Switzerland as a region of interest in order to leverage high-quality model input layers and applications to meet typical stakeholder needs.

Widely used vegetation indices and mechanistic land surface models are not effectively informed by the full information contained in Earth observation data and the observed spatial heterogeneity of land surface greenness responses at hillslope-scale resolution. Effective learning from the simultaneous evolution of climate and remotely sensed land surface properties is challenging. Modern deep learning and machine learning techniques, however, have the capacity to generate accurate predictions while also explaining the relationship between climate and its recent history, its position in the landscape, and its influences on vegetation. The task is to predict the future NDVI over forest areas to infer droughts, given past and future weather and surface reflectance. Giving future weather predictions as an input to the model, we are going for a 'guided prediction' approach where the aim is to exploit weather information from forecasting models in order to increase the predictive power of the model. Models are fully data-driven, without feature engineering, and trained on spatio-temporal data cubes, which can be seen as stacked satellite imagery for a specific geo-location and a timestep of past Sentinel 2 surface reflectance, past (observed) and future (forecasted) climate reanalysis, time-invariant information from a digital elevation model, and a land cover map. In the temporal domain, models are trained on the period between 2018-2019, validated between 05/2021 and 09/2021, and tested between 05/2020 and 09/2020.

In this research, we propose a methodology for how to successfully integrate future data from different modalities to go for a "guided-prediction" approach to enhance the predictive power of the models. We also propose a novel, complete guideline for how to effectively create earth observation data cubes. We conducted experiments regarding the model's performance under sparse conditions (clouds). We observed that the proposed model out-performed the baseline. However instead of learning the true signal, model "memorized" of the imputation values used to replace cloudy pixel values. We believe that the reasons for this are the small amount of data

to learn from, which effects the generalizability skill of the model, and our chosen cloud removal strategy.

# Contents


**1 Introduction** **1**

**2 Related Work** **5**

**3 Background** **7**
3.1 Drought Impact Definition . . . 7
3.2 Drought Impact Prediction . . . 8
3.3 Satellite Image Prediction . . . 9
3.3.1 Convolutional-LSTM Models . . . 9
3.4 Sequence Modelling with Encoder-Decoder Architecture . . . 12

**4 Data Creation and Processing** **15**
4.1 Data Sources . . . 16
4.1.1 Sentinel 2 . . . 16
4.1.2 ERA5: Global Reanalysis Model . . . 17
4.1.3 NASA SRTM Digital Elevation Model . . . 18
4.1.4 COPERNICUS Landcover v3 . . . 19
4.2 Exploratory Data Analysis . . . 20
4.3 Data Downloading . . . 24
4.4 Creating Datasets . . . 31
4.4.1 Data Sample Generation Pipeline . . . 32
4.5 Data Processing . . . 38

**5 Approach** **43**
5.1 Guided Prediction . . . 43
5.2 Drought Modelling . . . 46

**6 Experiments and Results** **49**
6.1 Experiments . . . 50
6.2 Results . . . 50

**7 Discussion** **55**
7.1 Data Operations . . . 55
7.2 Modelling . . . 57

**8 Conclusion and Future Work** **59**

**9 Appendix** **61**
9.1 Explanotary Data Analysis and Data Cleaning . . . . . 61
9.2 Data Downloading . . . . . 64
9.3 Dataset Creation . . . . . 66

# Chapter 1

# Introduction

Climate change is causing an increase in global temperatures, leading to changes in precipitation patterns and more frequent and severe droughts. As the Earth's temperature increases, so do evaporation rates, leading to drier soil and reduced water availability for plants. Additionally, warmer temperatures cause changes in atmospheric circulation patterns, leading to shifts in where precipitation occurs. This results in both an overall decrease in water availability and higher variability, making it harder for ecosystems and societies to adapt to the changes. Droughts are a natural occurrence; however, climate change has accelerated the rate at which they occur and made them more severe. This can lead to a range of negative consequences, including an increase in wildfire risk Mukherjee et al. (2018)and stress, as well as damage to plants in general. Recent hot and dry summers in Central Europe in 2018 and 2019 resulted in widespread canopy defoliation and tree mortality. In 2018, a majority of ecologically and economically vital tree species in the temperate forests of Austria, Germany, and Switzerland suffered from severe symptoms of drought stress, like widespread discoloration of leaves and premature leaf shedding. This extreme drought stress led to unprecedented levels of drought-induced tree mortality in various species throughout the region. Additionally, unexpectedly potent residual effects of the drought were detected in the following year, 2019 Schuldt et al. (2020).

To mitigate the impacts of droughts, it is crucial to model, understand, and predict the potential droughts. This allows for proactive measures to be taken to reduce the likelihood and severity of the event Rolnick et al. (2022). Predicting droughts as a natural hazard is a challenging task due to the varying spatial and temporal characteristics of climate and environmental factors. The impacts of droughts are dependent on the specific region and time, and there is no universally accepted definition of how to classify or quantify? these impacts. Moreover, the stochastic and complex nature of the environment and atmosphere makes it difficult for conventional mechanistic models to capture both spatial and temporal dependencies. The use of remote sensing data in this specific prediction task also presents challenges. Remote sensing data are Earth observation data which are continuously obtained from space-borne and airborne sensors, as well as some other data acquisition measurements. Challenges with these types of data come from the huge volume, diversity and complexity of the data collected. Each sensor provides its own unique spatial, temporal, and spectral resolution, creating a diverse range of data Ma et al. (2015). Multi-dimensionality of the remote sensing data can be seen in the Figure 1.1.

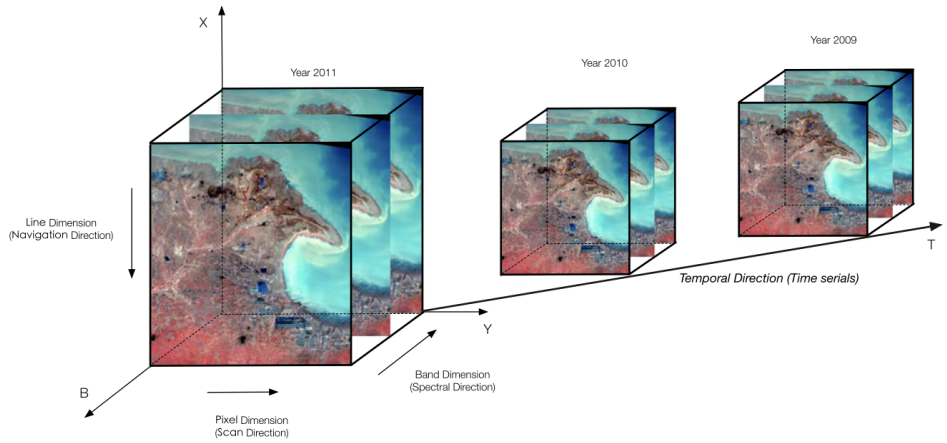



Figure 1.1: Multi-dimensional remote sensing data from Ma et al. (2015)

Because of the aforementioned problems, effective learning from the simultaneous evolution of climate and remotely sensed land surface properties is challenging. Modern deep learning and machine learning techniques however, have the potential to generate accurate predictions while also explaining the relationship between climate and other environmental variables, making them suitable for simulating and predicting natural hazards. The objective of this research is to train a model that predicts the Normalized Difference Vegetation Index (NDVI) values, which represent vegetation density and health, at the scene level. The proposed model utilizes historical NDVI data from preceding months, and future data from weather predictions as input to forecast NDVI to later infere on future drought impacts on forests in Switzerland. Data is obtained via IBM PAIRS geospatial data cluster Klein et al. (2015).

In this study, we present a methodology for predicting the impacts of drought utilizing a forward-looking approach. In the literature this approach is referred to as "future-conditioned" or "guided prediction" Kladny et al. (2022), Robin et al. (2022), Requena-Mesa et al. (2020). Deep neural networks are used as a method for this prediction task, and NDVI is used as a proxy for evaluating drought impacts. The region of interests are Jura, Schaffhausen, Vaud, and Valais which have been previously identified as affected by droughts in the research of Brun et al. (2020). The proposed model is fully data-driven, without feature engineering and is trained on spatio-temporal datacubes. The present research focuses on the modelling aspect such as the limitations and capabilities of neural networks in predicting NDVI to infere drought impacts as well as data processing aspect. The study aims to address the following research questions:

- **Research Question 1:** What are the optimal steps and methodologies to create earth observation data cubes that maximize the effectiveness of machine learning models?
- **Research Question 2:** How can we integrate future data from different modalities to enhance the predictive power of machine learning models?
- **Research Question 3:** How well do deep learning models predict NDVI under sparsity conditions (clouds)?

In chapter two, we provide an overview of the current state of the art methods in drought and natural hazard prediction. Chapter three describes the methods applied in this study. In chapter four, we present the pre-processing steps related to the preparation of a mutli-modal data set. In chapter five, we provide thorough explanations about the NDVI prediction task. In chapter

six and seven we present and discuss the results of our experiments and the performance of the proposed model. Finally, in chapters eight and nine we discuss and summarize the main findings and provide suggestions for future work.

# Chapter 2

# Related Work

Drought impact prediction can be formulated as a spatio-temporal satellite image prediction problem using satellite imagery as input and outputting future time-frames. Applications of satellite image prediction includes but is not limited to drought forecasting, crop-yield prediction, cloud image forecasting, and sea surface temperature prediction.

There are several ways to tackle satellite image prediction tasks using statistical and data-driven methods. Work from Rhee et al. (2016) focused on predicting drought indices for the upcoming six months in South Korea. The authors Rhee et al. (2016) used a Standard Precipitation Index as a proxy for defining droughts. Authors, Rhee et al. (2016) used machine learning models, including Decision Trees, Random Forests, and Extremely Randomized Trees, and compared their performance with regular interpolation methods. The final results showed that machine learning models outperformed interpolation methods for this task.

The work by Park et al. (2019) focused on drought area prediction in Korea without meteorological data. They propose a severe area drought prediction model (SDAP) that forecasts serious drought areas using Soil Moisture Index (SMI, Hunt et al. (2009)). The training area of the model was 7.5 km by 7.5 km and the model was trained with 65,000 samples. Features were mainly collected from LANDSAT 8 (Sturm et al. (2022)) and SRTM DEM (Jarvis et al. (2004)) data sources. Here, a random forest model was used. The work by Zhang et al. (2019) used XGBoost and Distributed lag nonlinear model (DLNM) for predicting the potential droughts quantified by the Standardized Precipitation Evapotranspiration Index, 1-6 months in advance. It was observed that XGBoost outperformed the neural network model in all experimental setups. Chiang and Tsai (2013) used a two-stage Support Vector Machine for predicting droughts. Here the authors took a classification-based approach, labeling the data as drought or not drought, feeding that into another SVM model as input and starting another classification process. Results showed that two-stage SVM out-performed the Bayesian classifier and the original SVM's.

Besides machine learning models, deep learning models have proven themselves with their capacity to approximate complex functions, such as environmental processes. Hence, they have become a preferred choice to learn patterns from high volumes of satellite imagery. Work by Shi et al. (2015) that focused on "precipitation nowcasting" proposes a novel approach to this task that is called 'Convolutional Long-Short-Term–Memory (ConvLSTM) models'. The drought forecasting problem is reformulated as a spatio-temporal sequence forecasting problem, in which both the input and the prediction target are spatio-temporal sequences. The primary driving force behind this method is to combine the advantages of both potent deep learning techniques, Long-Term-Short-Term-Memory models Sherstinsky (2018) and Convolutional Neural Networks O'Shea and Nash (2015a). LSTM networks are known for their capacity to handle sequential data and capture temporal dependencies. Convolutional Neural Networks are the backbone of any application that works with images because of their capability of learning from spatially correlated data. Combining these two approaches creates a great opportunity for handling spatio-temporal

time series satellite imagery. Experiments have been conducted with this approach and showed that ConvLSTM works better than conventional Fully-Connected LSTM networks Shi et al. (2015). There are numerous applications of Convolutional LSTM models that have resulted in success. Recent work by Robin et al. (2022), employed Convolutional LSTM models to forecast vegetation greenness in Africa. They formulated the land surface forecasting task as a guided video prediction task where the goal was to forecast the vegetation developing using topography and weather variables to guide the prediction. Experiments showed that ConvLSTM based models outperformed other baselines. In the work of Xiao et al. (2019), the goal was to predict the sea surface temperature field based on satellite imagery that is collected by the National Oceanic and Atmospheric Administration (NOAA). Petrou and Tian (2019), used Convolutional LSTMs to predict Sea Ice Motion. Another application is from Kladny et al. (2022) where the goal was to predict vegetation greenness given the EarthNet data cubes that are created from several satellite image data sources, Requena-Mesa et al. (2020). In both of these works, Xiao et al. (2019) and Kladny et al. (2022), Convolutional LSTM model was the best choice, outperforming LSTM models and U-Net's (Ronneberger et al. (2015)), respectively.

Work by Das and Ghosh (2016), which also takes a deep learning-based approach, proposes a new model architecture called Deep-STEP for Spatio-temporal prediction. The main objective of the approach proposed by Das and Ghosh (2016) is to reduce the complexity of the prediction task and improve scalability. Deep-STEP is based on deep stacking networks to increase the capacity of the overall architecture for modelling more complex functions. The proposed architecture is trained on LANDSAT satellite imagery for the United States between 2004-2010 and the prediction target was the NDVI in 2011 Pettorelli (2013). Deep-STEP was compared with Non-Linear Auto-Regressive Neural Networks, Multilayer Perceptron, and Deep Stack networks. Deep-STEP outperformed the compared methods in several evaluation metrics. Recent advances in Generative Adversarial Networks (GAN, Creswell et al. (2018) ) have made them a suitable choice for satellite image prediction. Work by Xu et al. (2019) uses a combination of LSTM networks and GANs to enhance the prediction capability of the model. The objective of this work was to predict cloud images of a future time frame. The model was tested on satellite cloud maps. The GAN-LSTM method did quite well in capturing the evaluation of the weather systems, and it outperformed traditional auto-encoder LSTM architecture.

Transformers, introduced by Vaswani et al. (2017) are also used for spatio-temporal satellite image prediction. The work by Huang et al. (2022), proposes a novel transformer-based model for multivariate time-series forecasting namely "Spatial-Temporal Convolutional Transformer Network (STCTN)". The main motivation behind the model is to create models that can capture context semantics while modeling temporal dependencies and to capture spatial dependencies of multiple patterns. Hence, they propose two novel attention mechanisms to capture both temporal and spatial dependencies. The model has been tested on different datasets. The results show that the STCTN was superior to existing methods which include Fully-Connected LSTM. To leverage Encoder-Decoder structures and convolution operations Hong et al. (2017) proposed a Convolutional Sequence-to-Sequence model for predicting undiscovered weather situations from previous satellite images. Experiments showed that the proposed model outperforms primitive LSTM architectures. As the literature review shows, there are numerous approaches to tackle time-series satellite image prediction. The choice of models is highly dependent on the task and data itself. The literature search showed that most widely used techniques, however, fall under the recurrent and convolutional neural network categories.

# Chapter 3

# Background

## 3.1 Drought Impact Definition

Drought prediction requires clearly defining the term and identifying key characteristics. As mentioned in the previous chapters, there is no universally accepted drought definition, and developing a singular definition of drought has been a persistent issue. This issue is due to the diverse range of existing drought concepts Dracup et al. (1980). To identify a drought, it is necessary to examine the environmental and spatial features of the region under consideration and apply an appropriate metric to quantify the concept of drought. Which metric to use is highly dependent on the drought type that one is willing to identify. There are three main drought types Hao et al. (2018):

- **Meteorological Drought:** Meteorological drought happens when there is a precipitation deficit. In this case one of the most common indices used as a proxy for droughts is *Standard Precipitation Index (SPI)* McKee et al. (1993). Negative SPI indicates drier than normal conditions and it identifies as drought. Another famous index is the *Palmer Drought Severity Index (PDSI)* Alley (1984). The PDSI is a standardized index based on a simplified soil water balance and estimates relative soil moisture conditions Rhee and Carbone (2007).

- **Agricultural Drought:** Agricultural Drought is related to a deficit in the soil moisture, which affects plant production and crop yield. Commonly used indices are: *Soil Moisture Percentile (SMP) Sheffield et al. (2004), Crop Moisture Index (CMI) Palmer (1968), and Soil Moisture Deficit Index (SMDI) Narasimhan and Srinivasan (2005)*. All of the the indices mentioned for identifying agricultural drought focus on quantifying soil moisture.

- **Hydrological Drought:** Hydrological drought is commonly associated with the shortage of surface runoff, streamflow, reservoir storage, or groundwater level. A commonly used hydrological drought index is *Palmer Hydrologic Drought Index Heim et al. (2000).*

In this work, we are focus on the Normalised Vegetation Difference Index(NDVI). NDVI is a commonly used vegetation index that is calculated from the reflectance values of the visible and near-infrared bands of remote sensing imagery. Equation for calculating is the following where "NIR" is near-infrared and "VIR" is the red band:

$$NDVI = (NIR - VIR)/(NIR + VIR) \tag{3.1}$$

NDVI values range from -1 to 1, with higher values indicating more dense and healthy vegetation. NDVI has become a widely used tool for monitoring vegetation health and assessing vegetation dynamics, particularly in areas such as agriculture, forestry, and ecological research. It has been

used as a proxy for detecting droughts in many previous works Nanzad et al. (2019),Brun et al. (2020), Sruthi and Aslam (2015),Peters et al. (2002). NDVI obtained from satellite data can be employed to identify rainfall shortages and evaluate meteorological and agricultural drought patterns in a timely and spatially explicit manner. Unexpected low NDVI values can indicate unexpected browning, which can lead to the impact of droughts Brun et al. (2020). Hence, in this work we decided to follow previous approaches and chose to use NDVI as a proxy for detecting droughts.

## 3.2 Drought Impact Prediction

NDVI anomaly is the simplest and most common NDVI-based approach for detecting and mapping drought using the long-term mean for a pixel or region at a given time Anyamba and Tucker (2012). There have been several studies regarding the use of NDVI as a proxy for drought identification in different regions. One of them comes from Nanzad et al. (2019). In this work, authors used NDVI anomalies for drought monitoring focusing on Mongolia for the years between 2000-2016. To derive the NDVI anomaly, first the mean NDVI of the growing season (May to September) for each year has been computed using the following formula:

$$NDVI_{mean_i} = (NDVI_1 + NDVI_2 + ... + NDVI_n)/n \tag{3.2}$$

where the $NDVI_{mean_i}$ is the mean NDVI value of the growing season of $i$ year. After calculating the means of every year, the overall mean of all NDVI means were computed with the following equation:

$$\overline{NDVI} = \sum_{i=1}^{n} \frac{NDVI_{mean_i}}{n} \tag{3.3}$$

where n is the number of years. The seasonally NDVI anomaly was then derived using the equation 3.3.

$$NDVI_{anomaly_i} = \frac{NDVI_{mean_i} - \overline{NDVI}}{\overline{NDVI}} \times 100 \tag{3.4}$$

Authors also provides an drought severity classification scheme that can be seen in 3.1.

| **Drought Severity Classes** | | | | |
|---|---|---|---|---|
| **Drought Level** | **Non-drought** | **Mild drought** | **Moderate drought** | **Severe drought** |
| **NDVI anomaly values** | Above 0 | 0 to -10 | -10 to -25 | -25 to -50 |

Table 3.1: Drought classification scheme according to Nanzad et al. (2019)

Besides the approach proposed by Nanzad et al. (2019), we also analyzed the approach proposed by Brun et al. (2020). The authors identified the impact of droughts on specific regions in Switzerland, including the Jura, Vaud, Valais, and Schaffhausen. They analyzed the change in NDVI differences in May/June between patches affected by early wilting and patches that were not affected during the 2018-2019 period They observed a difference of $-0.11$ in NDVI between affected and non-affected patches. Hence, this difference in NDVI can be used as an indicator of droughts. As we can see from the previous studies, there are various examples of how NDVI is used as a proxy for droughts. Our methodology utilises Brun et al. (2020)'s approach for detecting droughts, and it is covered in more detail in the Chapter 5.

## 3.3 Satellite Image Prediction

Satellite image prediction can be seen as a spatio-temporal time-series prediction problem. The problem presents several challenges, including high input dimensionality, high computation, and the chaotic and complex nature of the environmental and atmospheric variables to be modelled. In addition to capturing the temporal dependencies like in a regular time-series prediction task, one also has to capture the spatial dependencies, which adds another layer of complexity to the problem. Literature showed that combining several different model architectures for capturing both spatial and temporal dependencies worked best. In this work, one method to be used *Convolutional LSTMs*.

Convolutional LSTMs were first introduced by Shi et al. (2015) in order to nowcast precipitation. Shi et al. (2015), approaches the problem of precipitation nowcasting as a spatiotemporal sequence forecasting task, which can be addressed using the general sequence-to-sequence learning framework. To effectively capture spatiotemporal dependencies in the data, the authors extended the concept of an FC-LSTM model to a ConvLSTM model, which includes convolutional structures in both the input-to-state and state-to-state transitions.

### 3.3.1 Convolutional-LSTM Models

The Convolutional LSTM (ConvLSTM) model is a hybrid neural network that merges the Long-Short Term Memory (LSTM) architecture with the Convolutional Neural Network (CNN) in order to simultaneously analyze both temporal and spatial dependencies within a data set. The LSTM architecture is designed to process sequential data and to capture long and short-term dependencies by using gates to control the flow of information, while the CNN is designed to process image data and capture spatial dependencies through the use of convolutional filters. By combining these two approaches, ConvLSTM models are able to process data with both temporal and spatial dependencies, such as video sequences or satellite images, and to make predictions.

#### Long-Term Short Memory Networks

In an LSTM network, each memory cell in the network has three gates: an input gate, an output gate, and a forget gate. These gates control the flow of information into and out of the memory cell, allowing the network to remember and forget certain information over time. The input gate controls the flow of new information into the memory cell, the output gate controls the flow of information out of the memory cell, and the forget gate controls the removal of outdated information from the memory cell.

The mathematical components of an LSTM network are used to define the operations performed by the input, output, and forget gates, as well as the operations that update the memory cell state. These components are typically defined using matrix operations, and they are implemented using a combination of simple mathematical functions such as sigmoid, tanh, and element-wise multiplication Medsker and Jain (2001), Col (2015).

The input to the LSTM model is $\mathbf{X}_t \in \mathbb{R}^{n \times d}$ where $t$ is the timestep, $n$ is the batch size, and $d$ is the number of inputs. The hidden state of the previous time step is $\mathbf{H}_{t-1} \in \mathbb{R}^{n \times h}$ where $h$ is the number of hidden units. Weights and biases of the network are shown as $\mathbf{W}_{xi}, \mathbf{W}_{xf}, \mathbf{W}_{hc}, \mathbf{W}_{xo} \in \mathbb{R}^{d \times h}$ and $\mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_o \in \mathbb{R}^{1 \times h}$, respectively.

The basic mathematical components of an LSTM network are the following:

- **Input Gate:** The input gate determines which information from the current input should be added to the memory cell. It is defined using a sigmoid function, which outputs values between 0 and 1, indicating the degree to which each element of the input should be added

to the memory cell. The input gate is calculated as follows:

$$I_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i) \tag{3.5}$$

- **Forget gate:** The forget gate determines which information from the previous memory cell state should be retained in the current memory cell. It is also defined using a sigmoid function, with values closer to 1 indicating that more of the previous state should be retained.

$$F_t = \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f) \tag{3.6}$$

- **Output gate:** The output gate determines which information from the current memory cell should be outputted. It is defined using a sigmoid function, with values closer to 1 indicating that more of the current memory cell should be outputted.

$$O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o) \tag{3.7}$$

- **Memory cell update:** The memory cell update determines how the current input and previous state should be combined to create the new memory cell state. It is typically defined using a tanh function, which scales the result to the range [-1, 1], and an element-wise multiplication operation, which combines the input and previous state according to the input and forget gate values. The following equation is used to calculate the new candidate values for the cell state:

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{X}_t \mathbf{W}_{xc} + \mathbf{H}_{t-1} \mathbf{W}_{hc} + \mathbf{b}_c) \tag{3.8}$$

  In order to update the cell value, the old state is multiplied by the forget gate; this can be seen as forgetting some part of the information. Then the new candidate values are multiplied with the input gate, meaning how much we decided to update the state. Finally, the two values are added to get the new cell state.

$$\mathbf{C}_t = \mathbf{F}_t \odot \mathbf{C}_{t-1} + \mathbf{I}_t \odot \tilde{\mathbf{C}}_t \tag{3.9}$$

- **Hidden State:** Finally the hidden state is calculated via:

$$\mathbf{H}_t = \mathbf{O}_t \odot \tanh(\mathbf{C}_t) \tag{3.10}$$

The structure of an LSTM can be seen in Figure 3.1

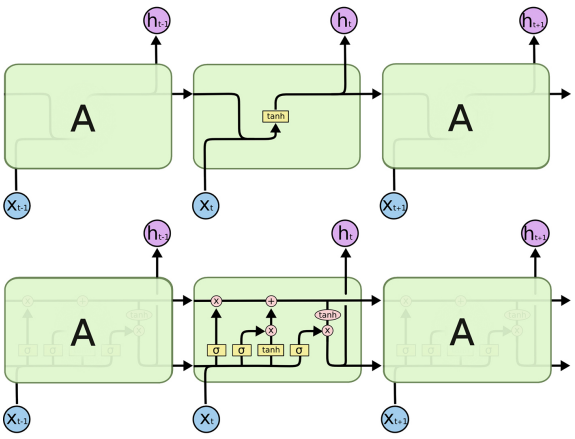



Figure 3.1: LSTM network structure Col (2015)

## Convolutional Neural Networks

In a CNN, the input data are processed through a series of convolutional layers, each of which applies a set of filters to the input data to extract features and reduce the dimensionality of the data. The filters are trained to recognize certain patterns or features in the data, and the output of the convolutional layers is a feature map that encodes the presence or absence of these patterns O'Shea and Nash (2015b). The feature maps are typically passed through a series of pooling layers, which downsample the data by taking the maximum or average value of a group of adjacent elements. This reduces the dimensionality of the data and helps to reduce overfitting. The output of the pooling layers is typically passed through a series of fully-connected (FC) layers, which perform classification or regression on the features extracted by the convolutional layers Albawi et al. (2017). The output of the FC layers is the final prediction made by the CNN. The structure of a CNN can be seen in Figure 3.2

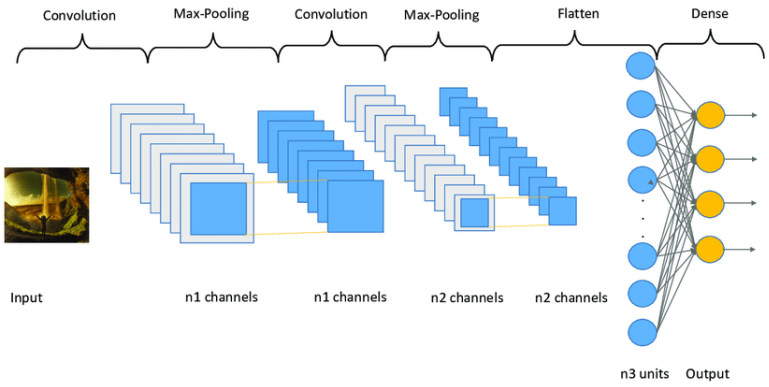



Figure 3.2: An example CNN architecture García-Ordás et al. (2020)

## ConvLSTM

A Convolutional LSTM (ConvLSTM) network can be seen as the best of both worlds. The idea is to combine the above-mentioned two architectures in order to capture both temporal and spatial dependencies. The change takes places in mathematical components of the LSTM. As mentioned above, LSTM models consist of internal matrix multiplications. However, in ConvLSTM models, internal matrix multiplications are replaced by convolution operations. Convolution and Hadamard product operations are shown with $*$ and $\odot$ respectively. Key equations of ConvLSTM are the following Shi et al. (2015):

- **Input gate:** $I_t = \sigma(X_t * W_{xi} + H_{t-1} * W_{hi} + b_i)$
- **Forget gate:** $F_t = \sigma(X_t * W_{xf} + H_{t-1} * W_{hf} + b_f)$
- **Output gate:** $O_t = \sigma(X_t * W_{xo} + H_{t-1} * W_{ho} + b_o)$
- **New candidate cell values :** $\tilde{C}_t = \tanh(X_t * W_{xc} + H_{t-1} * W_{hc} + b_c)$,
- **New cell state:** $C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t$
- **Hidden state:** $H_t = O_t \odot \tanh(C_t)$.

The Inner structure of ConvLSTM can bee seen in the Figure 3.3

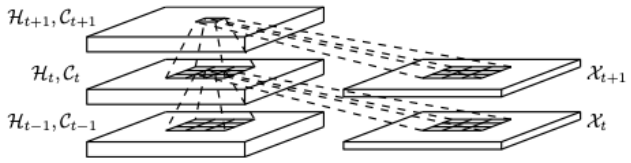



Figure 3.3: Inner structure of ConvLSTM, taken from Shi et al. (2015)

Convolutional LSTMs have been extensively employed as the primary computational architecture in several studies aiming to predict future satellite image data such as Kladny et al. (2022),Xiao et al. (2019), due to their ability to effectively capture spatio-temporal dependencies and incorporate convolutional features. The present work is built upon these earlier studies demonstrating the effectiveness of ConvLSTMs.

## 3.4 Sequence Modelling with Encoder-Decoder Architecture

Satellite image prediction can also be represented as a sequence-to-sequence modeling problem, in which the goal is to use the past n timestamps as input to predict m future timestamps (i.e, many-to-many or one-to-many prediction). One approach for many-to-many prediction is to use a combination of sequence-to-sequence (seq2seq) model architectures and Convolutional-Long Short-Term Memory (ConvLSTM) models. The effectiveness of this approach comes from the fact that seq2seq models are able to map input sequences to output sequences capturing temporal dependencies, while ConvLSTM models are able to capture local dependencies in the data using convolutional filters. By combining these two types of models, it is possible to achieve improved performance on tasks that involve predicting multiple timestamps based on multiple input timestamps. Work by Hong et al. (2017) demonstrated this approach for predicting weather conditions from previous satellite images. Hong et al. (2017) proposed to have an Encoder-Decoder framework using Recurrent Neural Networks. Each encoder and decoder section consists of several convolutional layers and ConvLSTM cells. In the encoder section of the seq2seq model, existing weather condition data are processed and encoded as a compact representation through time. The decoder then receives outputs and network states from the encoder and uses this information to sequentially predict future weather conditions. This approach allows the model to use the past weather data as input and generate a series of forecasts for future timestamps. The encoder and decoder work together to map the input sequence to the output sequence, allowing the model to effectively capture the dependencies and trends in the data. Thus, in this work we also go for a ConvLSTM autoencoder model.Convolutional LSTM autoencoders are therefore well-suited for tasks involving time series data, such as drought prediction. ConvLSTM autoencoders are a variant of autoencoders that use ConvLSTM cells, which are able to capture local dependencies in the data using convolutional filters within each ConvLSTM cell, in the encoder and decoder. A simple visiualization of ConvLSTM autoencoders can be seen in the Figure 3.4.

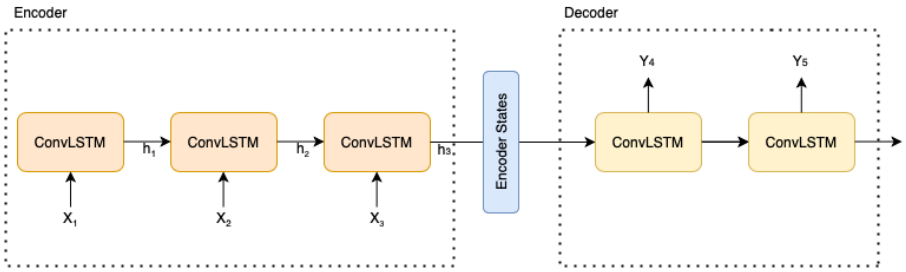



Figure 3.4: ConvLSTM Autoencoder

In this work we apply ConvLSTM Encoder-Decoder architecture which will be explained in more detail in the upcoming chapters.

# Chapter 4

# Data Creation and Processing

This work aims to study the impacts of the drought that occurred in Switzerland in 2018-2019 by focusing on specific regions of interest: Jura, Schaffhausen, Vaud, and Valais. The regions of interest can be seen in Figure 4.1 with purple representing Vaud, black representing Jura, green representing Valais, and light green representing Schaffhausen.The necessary Earth observation data for this analysis is obtained from IBM PAIRS Environmental Intelligence Suite, a geospatial big data service that contains a massive amount of curated geospatial data from various public and private sources. The data are stored in a unified format with the same Coordinate Reference System (CRS), which allows for easy integration of data from different sources. The PAIRS platform also allows for fast and efficient querying of large amounts of data.

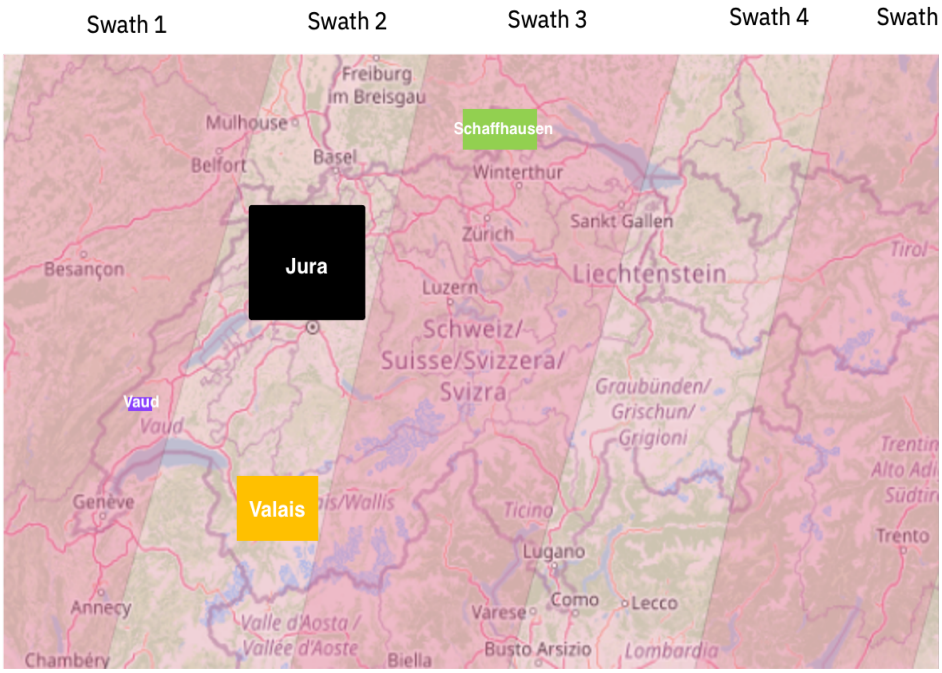



Figure 4.1: Regions of interest

## 4.1 Data Sources

IBM PAIRS geospatial big data analytics platform consists of many data sources that are stored in layers. Each data source has several layers that each contains different information. Four data sources to be used from IBM PAIRS were used : Sentinel 2 Drusch et al. (2012), ERA5 Hersbach et al. (2020), NASA SRTM Digital Elevation Model, and Land Cover from COPERNICUS V3 Copernicus (2016).

### 4.1.1 Sentinel 2

Sentinel-2, an Earth observation project from the Copernicus Program, routinely gathers multispectral optical imagery at high spatial resolution (10 m to 60 m) over land and coastal seas Drusch et al. (2012). An example of Sentinel 2 satellite imagery for the Jura region can be seen in 4.2. Sentinel-2A and Sentinel-2B are the two satellites that make up the mission's current constellation. Sentinel-2A launched in 2015 and Sentinel-2B launched in 2017. Sentinel-2 consists of visible, near infrared and shortwave infrared sensors comprising 13 spectral bands: 4 bands at 10 m, 6 bands at 20 m and 3 bands at 60 m spatial resolution ). The Sentinel 2 data product contains several data layers. An example data layer from Sentinel 2 can be seen in Figure 4.2. A list of data layers from Sentinel 2 and their meta information can be seen in Table 4.1

Figure 4.2: Sentinel 2 - Normalized Vegetation Index Data Layer for Jura region taken on 08-23-2016

The Sentinel 2 satellite imagery data, which typically has a resolution range of 10-60 meters, is stored in the IBM PAIRS platform at a resolution of 7 meters. This is achieved by using a bi-linear interpolation method.

| Data Layer | Raw Resolution | IBM PAIRS Resolution | Band | Wavelength (Sentinel 2 A/B) |
|---|---|---|---|---|
| Band 2 | 20 m | 7 m | Blue | 496.6/492.1 nm |
| Band 4 | | | Red | 664.5/665.0 nm |
| Band 8 | | | Near Infrared | 835.1/833.0 nm |
| Band 12 | 10 m | 7 m | SWIR | 2202.4/2185.7 nm |
| Band 11 | | | | 1613.7/1610.4 nm |
| Band 8a | | | Narrow Infrared | 864.8/864.0 nm |
| Band 7 | | | Vegetation Red Edge | 782.5/779.7 nm |
| Band 6 | | | | 740.2/739.1 nm |
| Band 5 | | | | 703.9/703.8 nm |
| Band 3 | | | Green | 560.0/559.0 nm |

Table 4.1: List of data layers from Sentinel 2

In addition to the data layers shown in Table 4.1, three additional data layers were also used. These are: **Scene Classification, Normalised Difference Vegetation Index, and Cloud Probability**. Scene classification data layer, provides pixel-by-pixel classification of an image using four types of clouds, cloud shadows, vegetation, soils/deserts, water and snow, and has a spatial resolution of **seven meters**. The normalized difference vegetation index (NDVI) data layer contains information about the level of vegetation present in each pixel. It is calculated by using Near Infrared and Red bands, which correspond to Sentinel 2 Bands 8 and 4 respectively. The formula for calculating NDVI is the following:

$$NDVI = (NIR - VIR)/(NIR + VIR) \tag{4.1}$$

### 4.1.2 ERA5: Global Reanalysis Model

ERA5 is a global climate reanalysis dataset that provides hourly estimates of a large number of atmospheric, land and oceanic climate variables with a spatial resolution of 30 km Hersbach et al. (2020). Climate reanalysis is a scientific method that combines historical weather observations with modern computer models to reconstruct a comprehensive picture of past weather and climate conditions, and a climate reanalysis dataset is a collection of weather and climate data that has been produced using the climate reanalysis process. ERA5 is one of the most famous climate reanalysis dataset with high temporal resolution. Data layers used from ERA5 are shown in Table 4.2

In addition to the raw data layers provided by IBM PAIRS, additional features were also calculated using "User Defined Functions" (UDF). In PAIRS, one can do mathematical operations on different layers, such as adding or subtracting layers, to create new features. Using user defined functions, we calculated Daily Max Vapour Pressure Deficit and Daily Min Vapour Pressure Deficit. Water vapor pressure deficit (WVPD) is a measure of the difference between the amount of water vapor in the air and the amount of water vapor that the air can hold when it is saturated, and it is often used as an indicator of how efficiently plants can transpire and uptake water from the soil. WVPD has been identified as an increasingly important driver for droughts Grossiord et al. (2020), and it is highly correlated with temperature and precipitation anomalies, that could result in drought Park Williams et al. (2013). Hence, it is an important feature to have. Max and min vapour pressure deficit variables are calculated via the following formula from the source Abtew et al. (2013) where $T_{max}$ and $T_{min}$ are the maximum and minimum temperature respectively. Relative humidity and temperature were obtained from the IBM PAIRS data.

- **Daily Max Vapour Pressure Deficit:**

  - *Max Saturation Water Vapor Pressure:* $SWVP_{max} = 611 \times \exp((17.27 \times T_{max})/(T_{max} + 237.3))$

  - *Max Vapour Pressure Deficit:* $VPD_{max} = SWVP_{max} \times (1 - relative\_hummidity_{min})$

- **Daily Min Vapour Pressure Deficit:**

  - *Min Saturation Water Vapor Pressure:* $SWVP_{min} = 611 \times \exp((17.27 \times T_{min})/(T_{min} + 237.3))$

  - *Min Vapour Pressure Deficit:* $VPD_{min} = SWVP_{min} \times (1 - relative\_hummidity_{min})$

In comparison to other data sources, the European Centre for Medium-Range Weather Forecasts (ECMWF) Re-Analysis 5 (ERA5) data has a relatively coarse resolution of 30 kilometers. However, in the IBM PAIRS platform, the ERA5 data are made available at a resolution of 11 kilometers through the use of a bi-linear interpolation method. A comprehensive summary of the available layers and their resolutions can be found in Table 4.2.

| **Data Layer** | **Raw Resolution** | **IBM PAIRS Resolution** |
|---|---|---|
| Total precipitation | 30 km | 14 km |
| Temperature | | |
| Atmospheric Water Vapor Content | | |
| Precipitation type | | |
| Solar radiation | | |
| Total cloud cover | | |
| Min temperature | | |
| Max temperature | | |
| Surface pressure | | |

Table 4.2: ERA5 Climate Reanalysis Model Data Layers and corresponding spatial resolutions

### 4.1.3 NASA SRTM Digital Elevation Model

NASA Shutter Radar Topography Mission provided a static digital elevation map with 30 m spatial resolution. The Digital Elevation Model for Jura can be seen in Figure 4.3

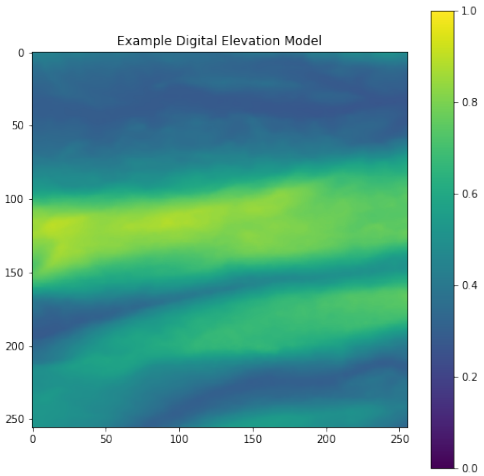



Figure 4.3: NASA SRTM Digital Elevation Model - for Jura region

### 4.1.4 COPERNICUS Landcover v3

Landcover v3 is a map that contains spatial information on different types (classes) of physical cov- erage of the Earth's surface, e.g. forests, grasslands, croplands, lakes, wetlands. Landcover v3 data is captured only once per year between 2015-2019. The spatial resolution of Landcover v3 classification maps is 100 m. Landcover classification map for Jura can be seen in Figure 4.4

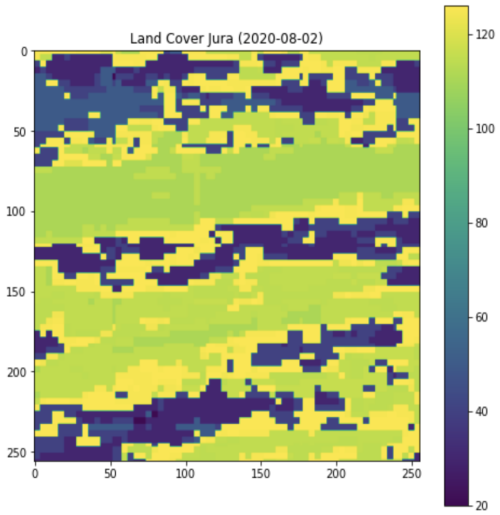



Figure 4.4: Landcover Classification Map - for Jura region on 2020-08-02

## 4.2 Exploratory Data Analysis

In this section, conducted an exploratory data analysis to gain a better understanding of the data available. This type of analysis allows us to examine the data in more detail, identify patterns, trends, and noise. As previously discussed, the Sentinel 2 constellation is composed of two satellites, Sentinel 2A and Sentinel 2B, which were launched at different times. Sentinel 2A was launched in 2015 and Sentinel 2B in 2017, resulting in an inconsistent temporal resolution for data collection. The temporal inconsistency between years can be seen in Figure 4.5.

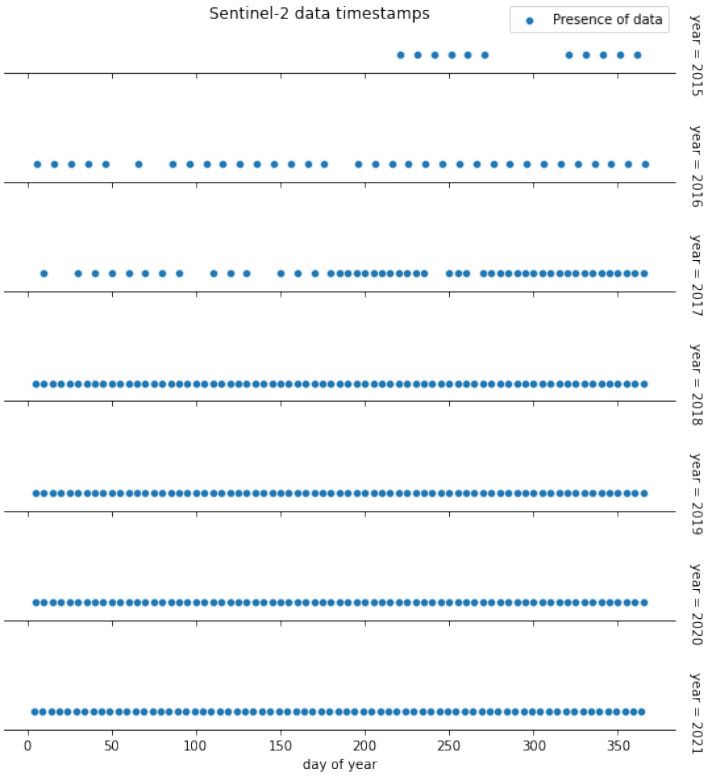



Figure 4.5: Temporal inconsistency between years in Sentinel 2

As expected, the frequency of data collection is lower prior to the middle of 2017 since only one satellite was orbiting. Following the analysis of temporal frequency at the annual level, a further analysis of the time series in different swath regions was conducted. In satellite remote sensing, a swath is the area on the Earth's surface that is covered by a single satellite sensor as it passes overhead. It is the portion of the Earth's surface that is imaged by the sensor during a single pass.The Sentinel-2 satellite imagery has a swath width of 290 kilometers, and Switzerland is covered by 5 distinct swaths. These swaths are illustrated in Figure 4.6.

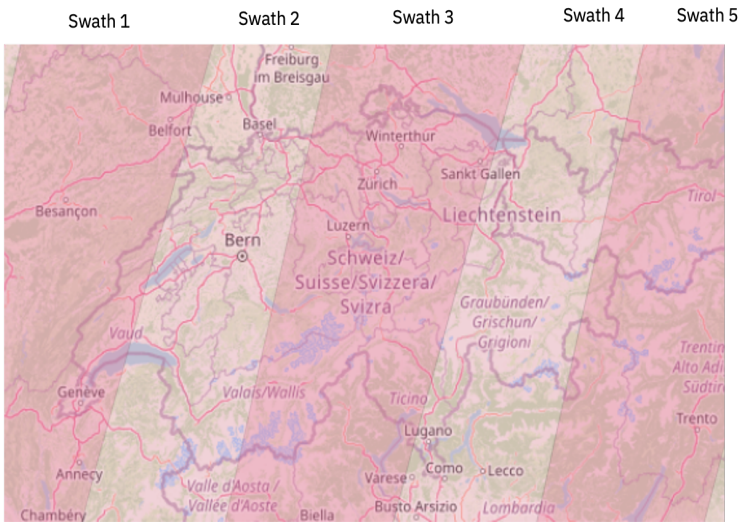



Figure 4.6: Sentinel 2 swaths in Switzerland

To identify return periods and any inconsistencies in the temporal domain across swaths, a point-wise data analysis was performed. The result of the analysis is illustrated in the Figure 4.7.

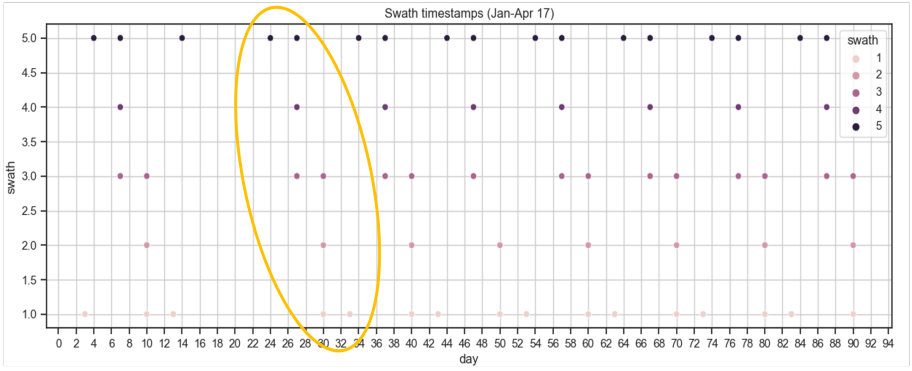



Figure 4.7: Time series irregularity between swaths in Switzerland. The yellow circle highlights the temporal inconsistencies between different swaths.

To conduct this analysis, a set of data points were retrieved from each swath region for the period between January and April 2017. The results indicate that variations in temporal frequency and date of image acquisition exist among different regions. The exposure to different swaths results in an irregularity in the time series of the data. For example, a region within Swath 1 exhibits a temporal resolution of ten days, while a region within Swath 5 exhibits a different temporal resolution. For example, for Jura region which is Swath 2, we observed 3-7 day revisiting times. Meaning, Sentinel 2 data was available in with a temporal resolution of three and seven days. This temporal irregularity among regions will cause a problems for both data processing and modelling. Temporal irregularity and its solution will be explained more in the upcoming sections.

After conducting an analysis of temporal coverage of the Sentinel 2 data, further examination on the NDVI layer of Sentinel 2 was performed. To investigate the trend of Normalized Difference Vegetation Index (NDVI) in the Jura region over the period from 2015 to 2021, we applied a seasonal decomposition technique. We extracted trend,noise, and seasonality from the signal to identify patterns. This is done using the **"statsmodels"** library in Python. The result can be seen in the Figure 4.8

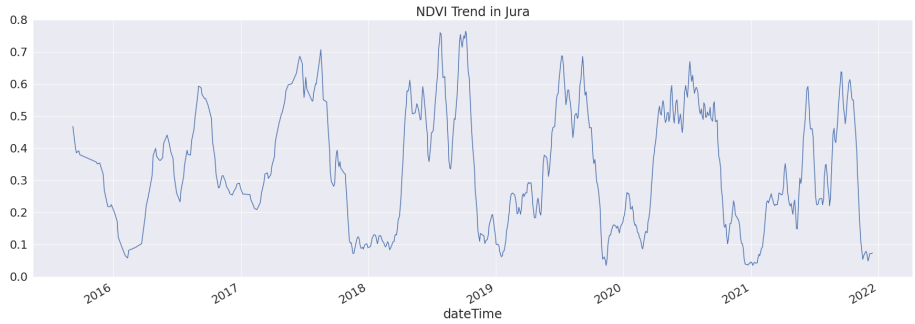



Figure 4.8: Time evolution of one pixel in Jura region from 2015 to 2022

As shown in Figure 4.8, the data exhibit a significant amount of noise. To confirm this observation, a pixel-wise analysis was conducted. Specifically, the temporal variation of NDVI values for a forest pixel located in the Jura region was examined. We analyzed the change of the NDVI values of a forest pixel between the interval 2015 and 2021. The result can be seen in Figure 4.9.

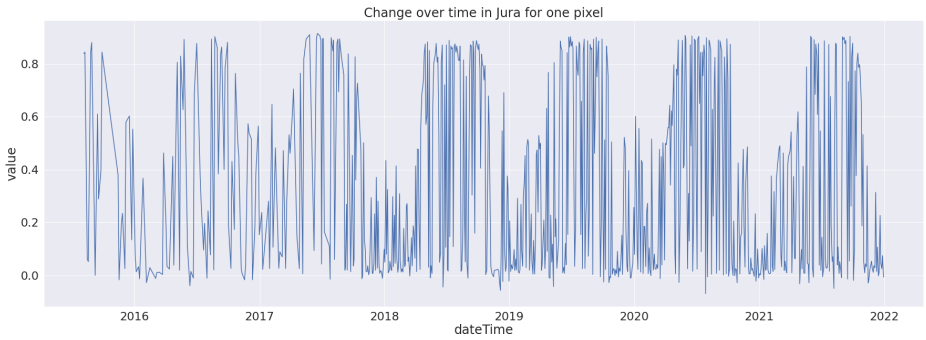



Figure 4.9: Change of NDVI value in one forest pixel

The outcome of the pixel-wise analysis supports our assertion that the data are significantly noisy. Our hypothesis is that a significant portion of the data's noise comes from the cloud coverage. Clouds add a significant amount of noise to the data and conceal the true light reflectance values of the pixels. When clouds are present in the satellite image, they can obscure the underlying vegetation, caus ing errors in the NDVI calculations. Clouds reflect sunlight, which can increase the amount of radiation in the red band and decrease the amount of radiation in

the infrared band, which is the opposite of what is expected in a vegetation-rich area. To verify this assumption, we used the Cloud Probability (CP) layer of Sentinel 2 to conduct a study on Jura region. We calculated means of NDVI for the summer months (May to September) and filtered the fully forested pixels with different levels of cloud probability thresholds. The aim of this analysis was to examine how different thresholds of cloud probability affect the NDVI data. The fully-forested pixels were dropped, if the cloud probability for that pixel exceeded a certain threshold. Results for the year 2015 can be seen in Figure 4.13. For more detailed results please see the appendix 9.3,9.2,9.1.

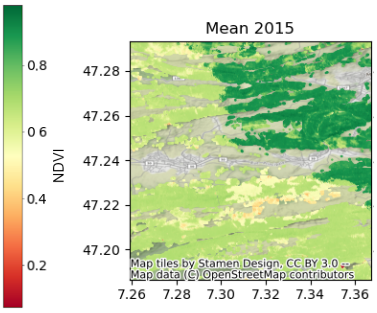



Figure 4.10: No filtering with CP

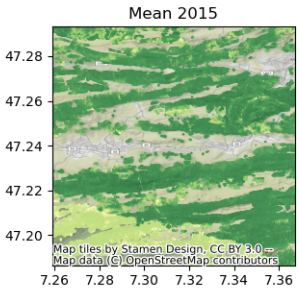



Figure 4.11: CP threshold: 70

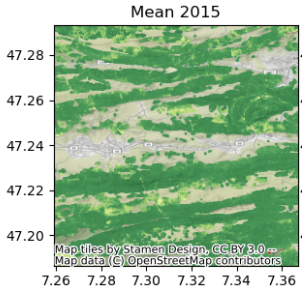



Figure 4.12: CP threshold: 30

Figure 4.13: The effect of different cloud probability thresholds on the NDVI data. Darker shades of green indicate higher NDVI values.

As can be seen in Figure 4.13, as we decrease the cloud probability threshold, or as we become more "strict" about clouds, there is an increase in the NDVI values. Hence, our hypothesis that the clouds cover the true value of the pixels and add noise is supported Pettorelli (2013),Rajitha et al. (2015). This significant amount of noise, would create problems in the modelling steps hence these data must be cleaned with care. More detail about the data cleaning procedure will be explained in the further sections.

## 4.3 Data Downloading

All the data was downloaded from the IBM PAIRS geospatial data cluster via API queries. The overall list of features and their spatial and temporal resolution information can be seen in the Table 4.3 As can be observed from Table 4.3, each data source has a different temporal and spatial

| Data Source | Band | IBM PAIRS Resolution | Temporal Resolution |
|---|---|---|---|
| Sentinel 2 | Red (Band 4) | ~ 7 m | ~ 5 days |
| | Green (Band 3) | | |
| | Blue (Band 2) | | |
| | NIR (Band 8) | | |
| | NDVI | | |
| | SCL | | |
| | SWIR (Band 11) | ~ 14 m | |
| | SWIR (Band 12) | | |
| | Vegetation Red Edge (Band 5) | | |
| | Vegetation Red Edge (Band 6) | | |
| | Vegetation Red Edge (Band 7) | | |
| | Narrow IR (Band 8a) | | |
| | Cloud Probability Mask | | |
| ERA 5 | Total Precipitation | ~ 14.5 km | Hourly |
| | Temperature | | |
| | Atm. water vapor content | | |
| | Precipitation Type | | |
| | Solar Radiation | | |
| | Total Cloud Cover | | |
| | Min Temperature | | |
| | Max Temperature | | |
| | Surface Pressure | | |
| | Max VPD | | |
| | Min VPD | | |
| WSL | Forest Mask | ~ 1 m | Static |
| Landcover Copernicus | Landcover Classification | ~ 56 m | Yearly (2015-2019) |
| NASA SRTM DEM | Digital Elevation Model | ~ 30 m | Static |

Table 4.3: List of data sources and features. The IBM PAIRS resolution is the interpolated resolution not the original spatial resolution of the bands.

resolutions. This variety adds another level of complexity to the modeling task. In order to model the data properly, all of the data sources should be aligned in both temporal and spatial resolution. Hence, data pre-processing steps are necessary. IBM PAIRS facilitates data pre-processing steps on the fly, meaning, one can apply basic data pre-processing before downloading the data without having to store the raw data locally. This feature of IBM PAIRS comes in handy when for large volumes of data, such as the **22 Peta-bytes** of Sentinel 2 data for Switzerland used for part of this project. All the IBM PAIRS data was obtained via queries to the PARIS API, with an IBM PAIRS query being similar to the following.

```
query_json = {
   "layers" : [
```

```
        {"type" : "raster", "id" : "49464"},
        "aggregation" : "Max"
    ],
    "spatial" : {
        "type" : "square",
        "coordinates" : ["35.6523", "-87.62", "51.6523", "-71.62"]
    },
    "temporal" : {"intervals" : [
        {"start" : "2018-09-01T00:00:00Z", "end" : "2018-10-01T00:00:00Z"}
    ]},
    "processor" : [{
        "order" : 1,
        "type" : "coarse-grain",
        "options" : [
            {"name" : "levelsUp", "value" : "2"},
            {"name" : "aggregation", "value" : "bilinear"}
        ]
    }]
}
query = paw.PAIRSQuery(query_json, PAIRS_SERVER, PAIRS_CREDENTIALS, authType='api-key')
query.submit()
```

Listing 4.1: Example IBM PAIRS query

Here, we are querying the data layer 49464 given by the **id** which is the NDVI layer from Sentinel 2, and we are querying the bounding box, defined in **spatial** part of the query, for one month of data as defined in the **temporal** section. One can also apply aggregation while downloading the data, in this example the aggregation method is taking the maximum. If needed, coarse-graining can also applied to the data while downloading. This operation reduces the resolution of the query result by a powers of 2 along each axis. Finally the output will be raster file as defined in **type**. In this study, all of the data was acquired using queries similar to the one demonstrated in this example. We fully utilized all of the PAIRS features.

As mentioned having different spatial resolutions would create an issue in the modelling task. To address this problem, we again used features of IBM PAIRS. One significant functionality of IBM PAIRS is that several data layers can be queried at once, and the resolution will be boosted to the data layer that has the highest resolution via the specified interpolation method. Here the interpolation method is *Bilinear*. By taking advantage of this aspect, Sentinel 2 and ERA5 data were queried together and the resolution was boosted to 7m. In order to match the resolutions of the static layers (DEM, LC Copernicus, and WSL Forest Mask), during the downloading process for every static layer a "dummy" Sentinel 2 layer was queried for each static layer just to match the resolutions. However, due to storage space issues, coarse-graining(down-scaling) had to be applied. The final resolution for all of the data sources was 28 m, which is the closest data resolution to the raw resolution of Sentinel 2 we could obtain with IBM PAIRS. In order to match the resolutions first we upscale them to 7m as previously mentioned and from there we apply down-scaling to go to 28m of resolution. If we had not upscaled the data first, for the ERA-5 data source, we couldn't have gone from 15 km to 28 m, since this kind of upscaling is not supported by IBM PAIRS.

In order to align data temporally, some aggregation operations were applied. We are interested in every timestep between 2015-2021 where Sentinel-2 is available; therefore, hourly ERA5 data had to be aggregated in such a way that it would match the temporal resolution of Sentinel-2 data. To determine which type of statistical aggregate to use for the ERA5 variables, the data

distribution for each variable of ERA5 was analyzed using quantile-quantile (QQ) Plots and histograms Das and Imon (2016). The QQ plot compares the distribution of the data to a theoretical distribution by plotting the quantiles of the data against the corresponding quantiles of the theoretical distribution Das and Imon (2016). It helps us to visually assess whether a given sample of data follows a specific probability distribution, which is the normal distribution in this case. Normally distributed data would not deviate too much from the quantile-quantile line. As for histograms, we divided the data to bins and superimposed a normal distribution on the histogram as a curve with the mean and standard deviation calculated from the data to see if our data follows a normal distribution. Normally distributed data would have a bell curved shape similar to the superimposed normal distribution curve. It is seen that none of the variables (total precipitation, NDVI, and solar radiation) follow a normal sistribution. QQ Plots and histograms for variables NDVI,total precipitation and solar radiation and can be seen in the Figure 4.17 and Figure 4.21 respectively. For complete list of QQ Plots and Histograms please the appendix 9.13, and 9.23

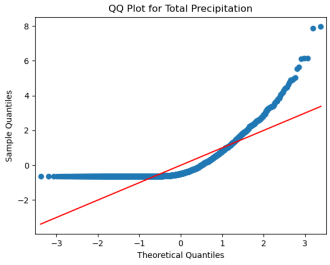



Figure 4.14: QQ Plot for Total Precipitation

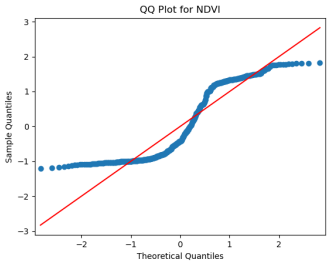



Figure 4.15: QQ Plot for NDVI

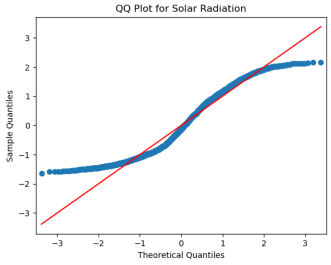



Figure 4.16: QQ Plot for Solar Radiation

Figure 4.17: Histograms for some of the ERA5 variables and NDVI Band from Sentinel 2.

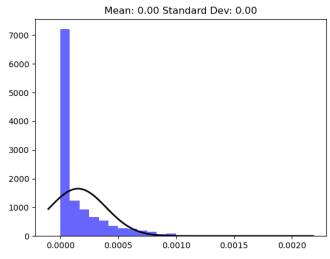



Figure 4.18: Histogram for Total Precipitation

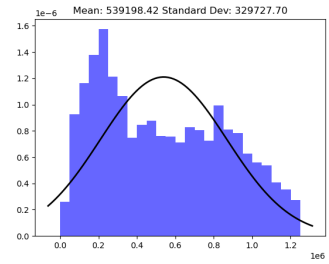



Figure 4.19: Histogram for Solar Radiation

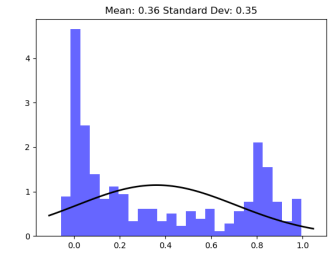



Figure 4.20: Histogram for NDVI

Figure 4.21: Histograms for some of the ERA5 variables and NDVI Band from Sentinel 2

In order to support the visual findings, the One Sample Kolmogorov-Smirnov Test (Berger and Zhou (2014)) was applied for each of the ERA5 variables. The Kolmogorov-Smirnov test is a non-parametric statistical test that compares the distribution of a sample to a reference proba-

| Variable | KS-Statistic | P-Value |
|---|---|---|
| Temperature | 0.039 | 1.335e-94 |
| Surface Pressure | 0.069 | 4.882e-11 |
| Min Temperature | 0.054 | 5.693e-07 |
| Max Temperature | 0.055 | 2.527e-07 |
| Atmospheric Water Vapor Content | 0.073 | 2.582e-12 |
| Solar Radiation | 0.098 | 6.152e-24 |
| Total Cloud Cover | 0.100 | 6.749e-23 |
| Total Precipitation | 0.263 | 4.430e-171 |

Table 4.4: One Sample Kolmogorov-Smirnov Test for ERA5 variables

bility distribution. The null hypothesis states that the sample comes from the same distribution as the reference.The alternative hypothesis states that the sample does not come from the same distribution as the reference distribution that one is interested. According to the test if the p-value is below some threshold (in this case it is chosen as 0.05) the Null Hypothesis is rejected. KS-Statistic tells us the goodness of fit, that can be seen as reject level for the hypothesis. Any value that is higher than 0.05 can be seen as as a good fit. In this case, we are testing whether our variables have a normal distribution. Determining if the data comes from the normal distribution or not will effect our decision about which aggregation to use. If the data comes from a normal distribution taking the mean would be a good aggregation methodology since this means that data values are centered around the mean, and it will be a good representation of the samples. If it does not come from a normal distribution, this means data is not uniform and might contain outliers. Hence, taking the median would be more fitting. As seen in the Table 4.4, results of Kolmogorov-Smirnov test supports our visual findings, concluding that none of the variables follows a normal distribution. These results suggest that using the median instead of the mean for aggregation would be more appropriate. However, the IBM PAIRS API does not support median aggregation. Therefore, an analysis was performed to determine how much the mean differed from the median. Thirty points around Switzerland were compared for three different variables of ERA5 to see the difference between the daily mean and daily median. Results of the analysis can be seen for some variables in the Figure 4.25.

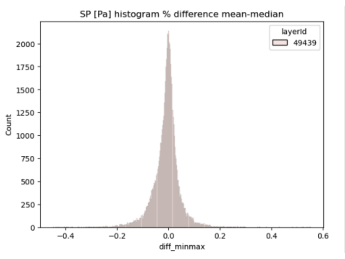



Figure 4.22: Mean and Median difference in percentage for Surface Pressure

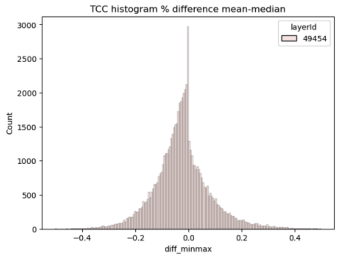



Figure 4.23: Mean and Median difference in percentage for Total Cloud Cover

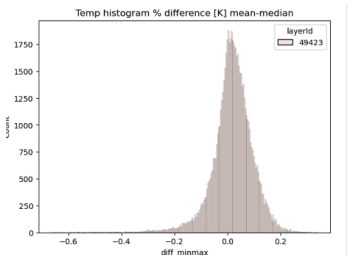



Figure 4.24: Mean and Median difference in percentage for Temperature

Figure 4.25: Difference between mean and median for some era variables.

As depicted in Figure 4.25 that the data are centered around the mean with deviation of ±20%

for surface pressure, and temperature and ±40% for total cloud cover. In order to see if the differences between means and medians are significantly different, the Wilcoxon Rank-Sum Test was conducted. The Wilcoxon Rank-Sum Test compares the medians of two groups by ranking the observations in each group and then comparing the rank sums. It compares the rank sums of the two groups and calculates a p-value to determine whether the medians of the two groups are statistically different. The p-value is the probability of observing the calculated rank sums or a more extreme rank sum under the null hypothesis, which is the hypothesis that the two groups come from the same distribution. If the p-value is less than a specified threshold (0.05), one can reject the null hypothesis and conclude that the medians of the two groups are statistically different. If the p-value is greater than the threshold (0.05), the null hypothesis is not rejected and the medians of the two groups are not considered statistically different. In this case, we compared a sample of data directly with its mean, to see if its median and mean are statistically different. This analysis was conducted only for three ERA variables which are Temperature, Surface Pressure, and Total Cloud Cover. Results of the Wilcoxon test showed that the difference between mean and median is not significant (p-values higher than 0.05), and since the data is centered around mean it has been decided to use mean instead of median, for some variables. We decided to use "min" and "max" aggregations for certain variables. Specifically, we track the highest and lowest values of temperature since high temperatures during the summer can lead to droughts. As for precipitation and solar radiation, we chose to take the sum since we believe that the total amount of precipitation and solar radiation in a region plays a significant role in determining its susceptibility to droughts Qi et al. (2022). Complete list of aggregations can be seen in the Table 4.5

| **ERA5 Features** | **Aggregation** |
|---|---|
| Temperature | Mean |
| Surface Pressure | Mean |
| Min Temperature | Min |
| Max Temperature | Max |
| Atmospheric Water Vapor Content | Mean |
| Solar Radiation | Sum |
| Total Cloud Cover | Mean |
| Total Precipitation | Sum |
| VPD Min | Min |
| VPD Max | Max |

Table 4.5: Features and aggregation methods used for ERA5 variables

## Data Downloading: Configuration A

In the first set up, only data from the Jura region was downloaded. Downloading ERA5 data could be conducted only when Sentinel-2 was available, and only values of the corresponding date were taken into account and aggregated. A flowchart that shows the data downloading logic can be seen in Figure 4.26

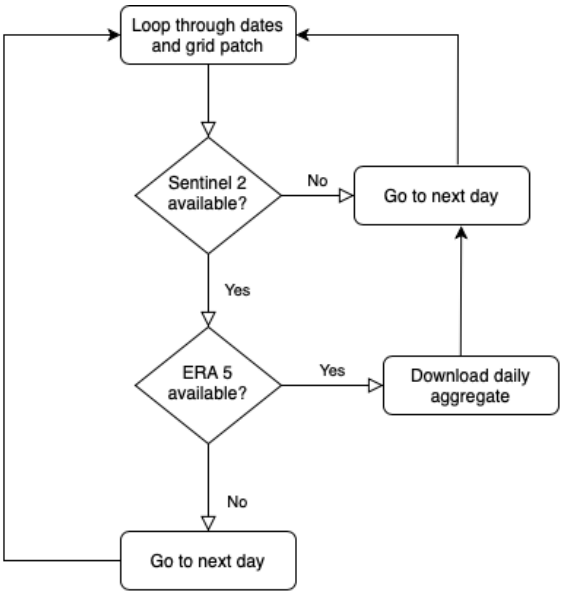



Figure 4.26: Data Downloading Scheme, configuration 1

## Data Downloading: Configuration B

For simplicity, while downloading the Jura region, ERA 5 data was only downloaded when Sentinel 2 was available, however in this configuration in order to benefit all the data that is available we go for a fairly different approach. For the regions Schaffhausen, Valais, and Vaud, the data download logic is similar to the configuration A, however, in this set up instead of taking into account only the matching dates between Sentinel 2 and ERA5, we considered all of the preceding days of ERA5 data and obtained an aggregate.The logic of this configuration is to download all of the ERA5 data between the current and last available dates for Sentinel 2. As mentioned in section. As mentioned in section 4.2 Switzerland is exposed to five different swaths, and each of these swath regions have different temporal resolutions. For the sake of simplicity only the regions that have similar temporal characteristics were downloaded. We have observed that first three swath regions obtain similar temporal behaviour as can be seen in Figure 4.27. While choosing which regions to work on, besides trying to chose where we know there were impacts of droughts via Brun et al. (2020), we also looked closely at the regions in swaths that have similar temporal characteristics. Chosen regions and their corresponding swaths are in Figure 4.28.


Figure 4.27: Swath regions 1,2,3 are showing similar behaviour

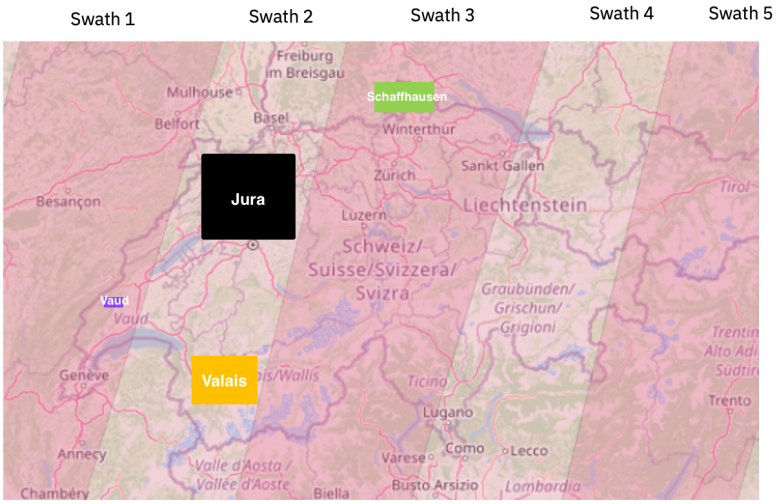



Figure 4.28: Swath regions and region of interests

Choosing temporally similar regions helps us with data aggregation. As previously stated, Sentinel 2-A and Sentinel 2-B satellites started to orbiting in different different times. This creates an irregularity in the time-series as can be seen in Figure 4.5. Hence, before 2017, ERA5 is aggregated to 10 days, as only 2A is orbiting, and after 2017 data is aggregated to 5 days,while downloading. However, after downloading data is aggregated to 10 days for simplicity.

Once the download is complete, total timestamps per region can be seen in the Table 4.6.

| Timestamps | Jura | Schaffhausen | Valais | Vaud |
|---|---|---|---|---|
| 2015 | 11 | 11 | 11 | 11 |
| 2016 | 34 | 33 | 34 | 34 |
| 2017 | 49 | 43 | 50 | 49 |
| 2018 | 73 | 69 | 71 | 72 |
| 2019 | 73 | 66 | 72 | 70 |
| 2020 | 73 | 67 | 52 | 73 |
| 2021 | 73 | 72 | 64 | 71 |
| Total number of timestamps | 386 | 361 | 354 | 380 |

Table 4.6: Number of timestamps

The difference between the number of timestamps before and after 2017 comes from the fact that Sentinel 2A and Sentinel 2B started orbiting at different times as mentioned above. Downloaded data files are in **.tiff** format, where each data layer is a separate file One tiff file consists of $1960 \times 1960$ pixels, with coordinate reference system **EPSG:4326**.

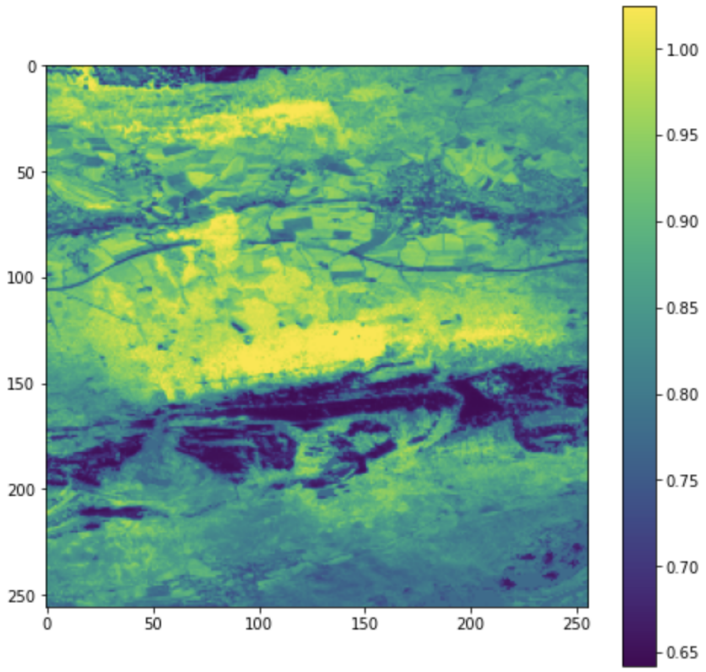



Figure 4.29: Example downloaded NDVI Layer

## 4.4 Creating Datasets

Our data sets consists of small data cubes. An Earth observation data cube is a data structure that is used to store and analyze remotely sensed data, such as satellite imagery. It is called a "cube" because it is typically multi-dimensional, with the dimensions representing the time,

channels, height and width. These data cubes are mostly represented by multi-dimensional arrays i.e. tensors, and they are created via stacking the corresponding satellite imagery by location and timestamp. One example illustration of a data cube for the Jura region can be seen in Figure 4.30. In this work, a unique data set for drought impact has been created, along with an exclusive data sampler to create "data cubes". Since current tools such as PyTorch Paszke et al. (2019) or TorchGeo Stewart et al. (2022) do not support data sampling both in temporal and spatial dimensions, both data set and data sampler modules of the PyTorch framework had to be tailored. In this section, we will explain the process of generating one data sample and constructing our training, testing, and validation sets based on it.

### 4.4.1 Data Sample Generation Pipeline

The goal here is to create data cubes for each region and time of interest, to sample smaller data cubes from it to generate datacset. An illustration of this task can be seen in Figure 4.30.

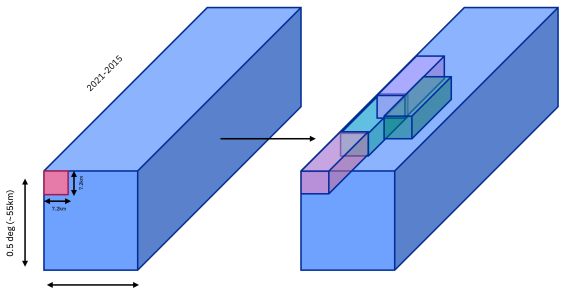



Figure 4.30: Creating data cubes for the region and time of interest (Left) and sampling of smaller cubes from it (Right)

To create the data cubes and samples, three components of the system work together. These are dataset class, data sampler, and data loader.

- **Dataset Class:** The objective of the dataset class is to generate a list of dates in a time series for a location given the input from the user.
- **Data Sampler:** Data sampler returns a bounding box, and an id for a sample.
- **Data Loader:** Data loader combines the information coming from dataset class and data sampler in order to load time series satellite imagery data within the bounding box that comes from the sampler.

#### Drought Impact Dataset

A specific data set for drought impact is created using the PyTorch framework. The PyTorch data set module can be used to manipulate data in an easy and efficient way, and it supports the deep learning pipeline. The objective of the drought impact data set is to fetch the satellite imagery of different data sources for a user-defined time range and region of interest. The flexibility of use of the data set was a key consideration when creating the dataset class. Data set creation must be as flexible as possible so that one could create a data set with different features and different times, different regions for experiment purposes. As input, dataset class takes the following arguments:

- **s2_paths**: Path to Sentinel 2 data.
- **era_paths**: Path to ERA5 data.
- **lc_paths**: Path to Landcover Copernicus data.
- **dem_paths**: Path to Digital Elevation Model.
- **data_file_extension**: Type of files being used
- **keep_out_list**: List of "regions" (family of filenames) to NOT include in data set (coordinates).
- **focus_list**: Only files starting with provided strings are included in the data set.
- **focus_time**: Start and end timestamps to be filtered.
- **ts_delta**: Number of days between each timestamp of the time series.
- **ts_len**: Total number of timestamps in the generated time series.
- **ratio**: Ratio used as condition on whether to use timeseries or not.
- **len_preds**: Number of timestamps that will be forecasted (necessary for stacking weather forecasts). Default is 1.
- **nan_handling**: Nan handling protocol within the image.
- **feature_set**: Set of features to include in the model.
- **agg_time**: Flag to indicate whether data will be aggregated or not.
- **norm_stat_1**: Normalisation statistic
- **norm_stat_2**: Normalisation statistic
- **multiple_labels**: Flag indicating if there will be multiple labels to predict.
- **correct_ndvi**: Cloud probability threshold to smoothen the data

**"DroughtImpactDataset"** class inherits from Dataset class of the Pytorch framework, and it is implemented as a "map-style" dataset. A "map-style" dataset allows you to access its data elements using an indexing-style syntax, like $dataset[i]$, where $i$ is the index of the data element. Class diagram of the dataset class can be found in Appendix 9.24.

How dataset class operates is as follows. First, we do a spatial filtering step, in which only data files for the region of interest are kept, and data files for other locations are filtered out. After obtaining the raw data files for the region of interest, we move on to temporal filtering. As an input, dataset class takes an **"focus_time"**, and **"ts_delta"** arguments as mentioned. Here the user can define the time range that they want to focus on, and the timestep ($\Delta T$) to be used. One example use could be for creating the training set. To build a training set for the time period 2018-2020 with a $\Delta T$ of ten days, one would define it while creating the dataset instance. In the temporal filtering step, dataset class takes this information from the user and generates all possible time-series. An illustration of the temporal filtering step can be seen in Figure 4.31.

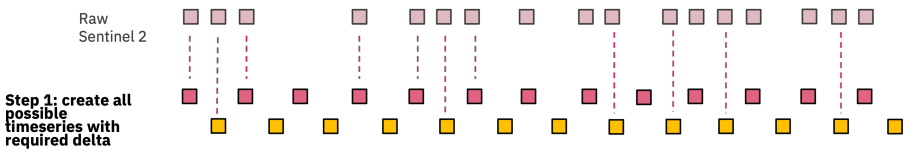



Figure 4.31: Dataset Class: Step 1. Given the time range and $\Delta T$, dataset class first generates all the possible time series that can be created.

After generating all possible timeseries, the next step is to split the timeseries in to sub-timeseries to define historical dates and forecasting dates. Within each sample, we have historical dates, meaning data that the model will use to "learn", and forecasting days (which can be thought of as labels). Historical and forecasting dates can also be seen as context and target in a data sample. In order to create these, dataset class uses the user defined variables: **"ts_len"** and **"ts_preds"**. Using this input, dataset class generates multiple timeseries that fits these criterias. For example, , it is possible to create a training set for between 2018 and 2019, with a $\Delta T$ of ten days, using three timestamps from the past ("ts_len") to forecast one timestamp ("ts_preds"). To do this, dataset class would produce a list of all of the possible time series fitting these criteria for the given region of interest. Each data sample would have different timestamps but adhere to the user-defined criteria. An illustration for this task can be seen in Figure 4.32.

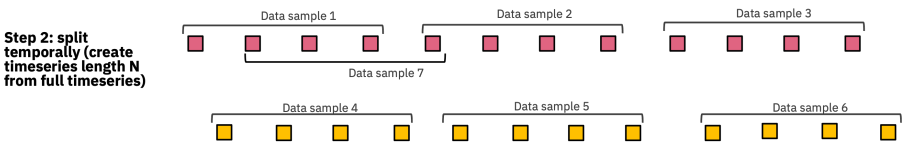



Figure 4.32: Dataset Class: Step 2

After the data are spatially and temporally filtered and split into sub-time series, dataset class finds the corresponding data files with these specifications. These data files are then read in an multi-dimensional array format and stacked together to create one big multi-dimensional (4-D) array or tensor , a.k.a a data cube for the whole region of interest. This resulting structure is illustrated in Figure 4.33.

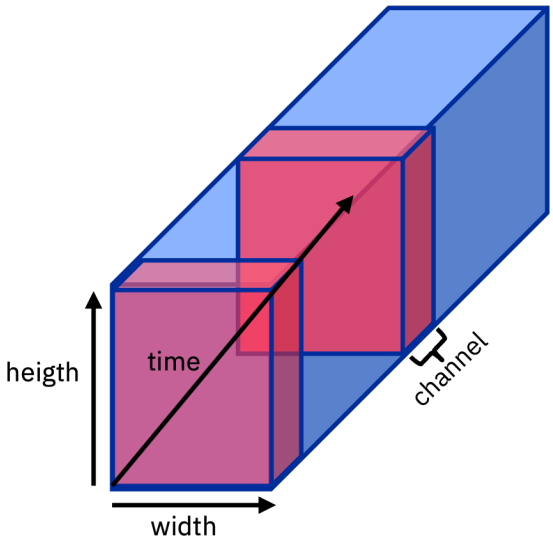



Figure 4.33: Data structure produced by dataset class. The four dimensions of the data structure are [time,channels,height,width]

In the end dataset class returns list of dates in a time series with the corresponding raster bounds for the region of interest. This output then be used by the Data Sampler.

### Drought Impact Data Sampler

Data sampler is an object that provides a way to randomly sample elements from a dataset. Here the objective is to sample the data both temporally and spatially. However, time series sampling is not supported by existing frameworks like PyTorch and TorchGeo. The data sampler is therefore tailored to meet the needs. As input, data sampler takes the following arguments:

- **dataset**: Dataset to sample from
- **size**: Size of each sample (number of pixels)
- **length**: Number of samples to generate
- **replacement**: Sample with or without replacement
- **Region of Interest**: Region of Interest to sample from
- **Mask**: File path to mask to use
- **Mask threshold**: How much should sample must contain the mask

**"DroughtImpactSampler"** class inherits from the Sampler class of the Pytorch framework. It uses an helper class called **"BoundingBox"**, and it contains methodologies to manipulate and apply operations on bounding boxes. Class diagram of the data sampler class can be found in Appendix 9.25. Data sampler takes the dataset class, and samples a random bounding box within the given raster bounds. Here the data sampler does not allow to sampling the exact bounding box, however we allow spatial overlap. Data sampler returns a bounding box for a sampled time-series, and an sample id. As an example of use of the data sampler, Bob, wants to generate

ten thousand data samples ("length" variable) from his dataset, with each sample having a size of $[256, 256]$ ("size" variable in pixels), from his region of interest. He also only wants to focus on forests in his data, so he provides the data path to his mask (mask variable) let's say "forest", and he wants that each of his data samples to contain at least 60% forest ("threshold" variable). Given these user-defined variables, data sampler would return random bounding boxes (but not overlapping completely) that match these criteria and sample ids for each sample. An illustration of what data sampler returns can be seen in Figure 4.34. The bounding boxes can then used with the information from dataset class as input for Data Loader

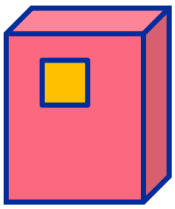


Figure 4.34: Yellow bounding box is returned by the data sampler

Finally, the data loader combines the information returned by the dataset class and data sampler. It gets the bounding box, and time series and loads the data sample. Example visualization of a data sample can be seen in the Figure 4.35. As mentioned before one data sample consists of context and target days where context is the historical data to be learned from and target is the days to be forecasted.

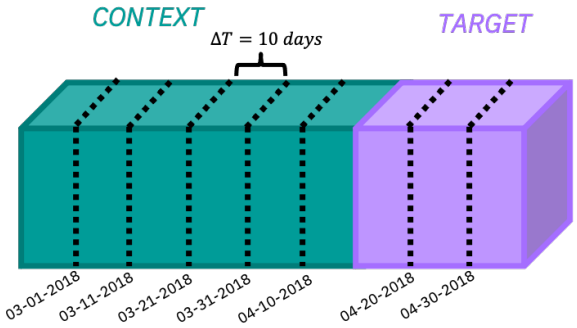



Figure 4.35: One data sample showing 6 historical (context) dates and 3 forecast (target) dates

The final data sample is a tensor with four dimensions that are: number of timesteps, number of channels, height and width. While creating the data samples for training, testing and validation sets, temporal overlap is allowed. Meaning, a target date in one sample can be a context date of another sample. However, overlap between training and testing sets is not allowed in order to avoid the problem of temporal leakage.

In conclusion, we implemented three main components in order to create our data samples for modelling: dataset class, data sampler, and data loader. Dataset class and Data Sampler had be tailored because existing frameworks were not fulfilling the needs of this research. A final sketch that shows how these three components interact with each other in a broader view can be seen in

Figure 4.36. For experiments training,testing, and validation sets are created with this framework. For trainin we are considering the time range **2018-2019**, for validation **05-2021 / 09-2021** (summer months of 2021), and for testing **05-2020 / 09-2020** (summer months of 2020). All the data created in a way that it would contain **49%** forests.

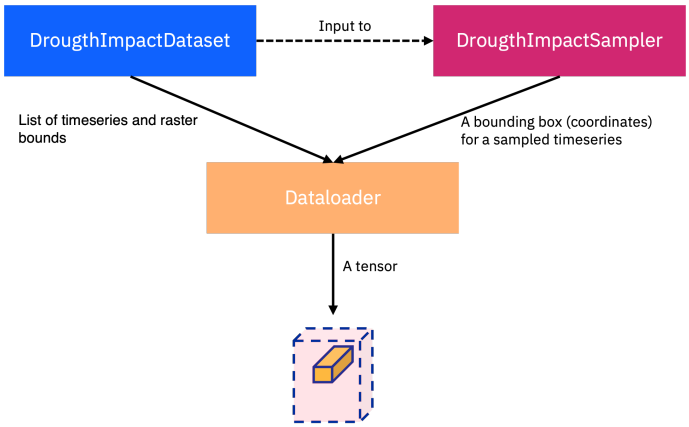



Figure 4.36: Overall data set creation pipeline

## 4.5 Data Processing

This section covers the data processing techniques applied to the data after it was downloaded.

As mentioned in section 4.2, in satellite images clouds introduce artefacts that interfere with data analysis, thus areas with clouds need to be omitted. Our cloud removal strategy is as follows: Generate a clean and smoothed NDVI signal with as few clouds as possible, then define a proper cloud probability threshold to remove the "cloudy" pixels and impute them with the cleaned signal. To generate the clean NDVI signal, we focused on data from the Valais region. The reasons why the analysis was conducted on the Valais data are that we know of significant effects of droughts in this region Brun et al. (2020), and that Valais is considered as a less cloudy region by MeteoSwiss. Hence, in the Valais region, it is easier to obtain a clear NDVI signal to observe the effects of clouds without being affected by noise too much. We took into account only cloud free forest pixels and calculate the median for each day. After calculating the median, we binned the data by week and dropped the lowest 5%, since it can still be affected by the clouds. To smoothen the data, we applied a Savitzky-Golay filter. The Savitzky-Golay filter is a signal processing technique that is used for smoothing noisy data while preserving important features of the signal. It is heavily used in environmental sciences because environmental data are often noisy and require smoothing. The resulting signal can be seen in Figure 4.39. In order to have values for every day, a linear interpolation method was applied on top of the Savitzky-Golay filter. The resulting signal can be seen in Figure 4.38.

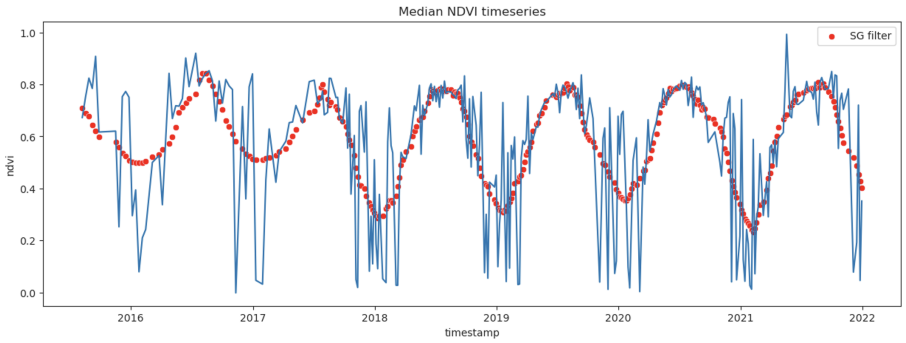



Figure 4.37: Savitzky-Golay applied signal

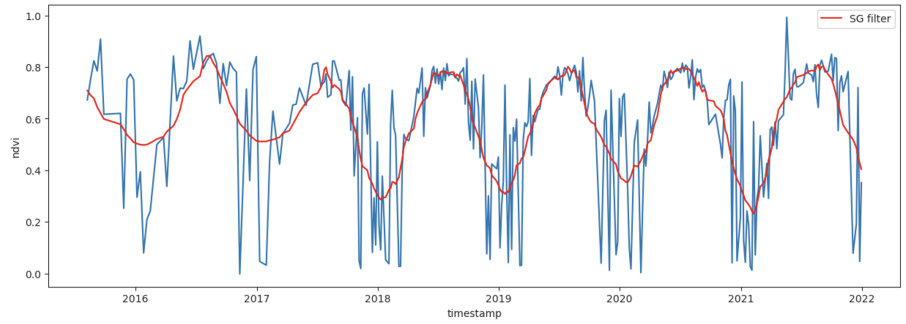



Figure 4.38: Linearly interpolated signal after applying Savitzky-Golay filter

Using this filtered signal, the goal was to impute the cloudy pixels in our data set. To do this, it was necessary to decide which cloud probability threshold to use. To find out the optimal cloud

probability threshold, we conducted an analysis of how cloud probability affects the mean and median NDVI values in Valais, Switzerland. wanted to observe the change in NDVI values under different cloud probability thresholds. For a selection of cloud probability thresholds, mean and median NDVI for pixels in summer images were calculated. As can be seen in Figure 4.39, NDVI is highlu sensitive to clouds. Increased cloud probability resulted in a big drop in the calculated NDVI values.

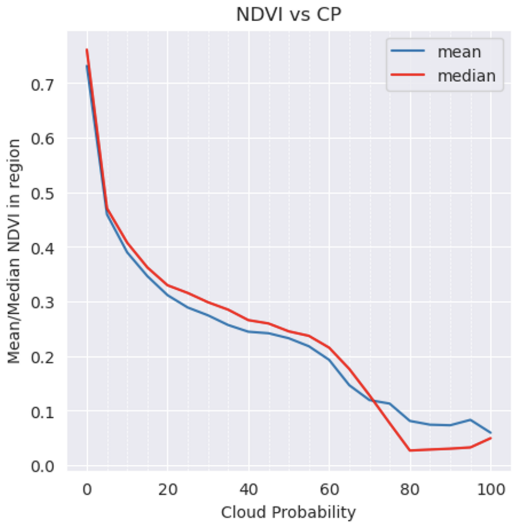



Figure 4.39: NDVI vs. Cloud Probability thresholds

Another important factor while determining what cloud probability threshold to apply is the data loss that occurs when "cloudy" pixels above a certain threshold are removed. An analysis was done to determine the data loss that would occur if we filtered with a selected threshold. Results can be seen in Figures 4.41, and 4.40. From 4.39 we can see that even if we choose a cloud probability threshold of 20%, to be strict with removal of clouds, we still retain more than half of the data. However, when we look at each year separately, we can see that not every year is affected the same. This is because not every year has the same number of data-points. There is a trade-off in deciding which cloud probability threshold to use. If one sets the threshold to be high, then the data would contain high noise. However, if one decides to set the threshold too low, then most of the values will be imputed, and we might lose sight of the real signal. In order to see what the resulting signal would be like, we decided to go with a threshold of 30%. Hence, we would remove every pixel that has more than 30% of cloud cover, and impute them with the "cleaned" signal we extracted. For the pixel that is removed, we get the corresponding pixel in the signal, and use that value to do imputation. The raw signal and the cleaned signal can be seen in Figures 4.42, and 4.43 respectively. In Figure 4.42, the scatter points corresponds to the raw signal, and the points that are connected with a red line are the pixels that passed the filter. After removing the pixels that are below the threshold, they have been imputed with the "cleaned" signal, and the corrected signal can be seen in 4.43. As we can see, the cleaned signal is not perfect. We can still see some artefacts in the data, here its assumed that artefacts come from cloud shadows and other atmospheric effects. However, no further analysis was made regarding to this assumption. A cloudy NDVI scene, its corresponding cloud probability mask, and resulting cleaned NDVI can be seen in Figure 4.44. A decision was made to set the cloud probability threshold at 30% in order

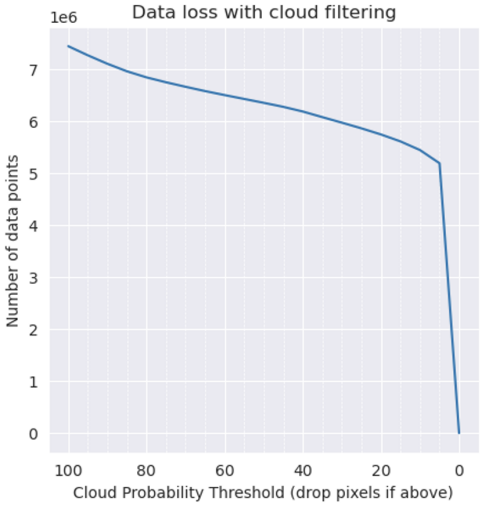



Figure 4.40: Mean/Median NDVI vs. Cloud Probability threshold


Figure 4.41: Data Loss vs. Cloud Probability

to strike a balance between maintaining data completeness and achieving data cleanliness.

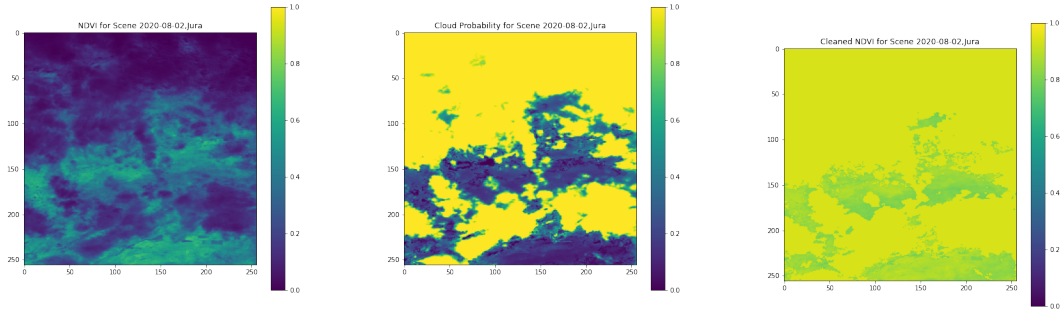



Figure 4.44: Cloudy pixels in the NDVI scene (Left), the cloud probability mask used to filter and impute the cleaned signal (Middle), and the resulting corrected NDVI scene (Right) in Jura region on 2018-04-05

Besides cleaning the data, another data processing step was dealing with missing values. Missing values can be the missing values within an scene (image), and they can also be missing days or missing a channel for a day. For handling missing values within a scene, the current framework supports filling the missing values via substituting with the mean, min, max, or via user-defined value. If a day is missing, the last available day is replicated for the corresponding days and bands. The last step is the normalisation of the data. Here min-max normalisation is used. Minimum and maximum values for each band across time were calculated and the data were normalised. Each data sample (scene) that has been created has a size of $[256, 256]$ pixels,

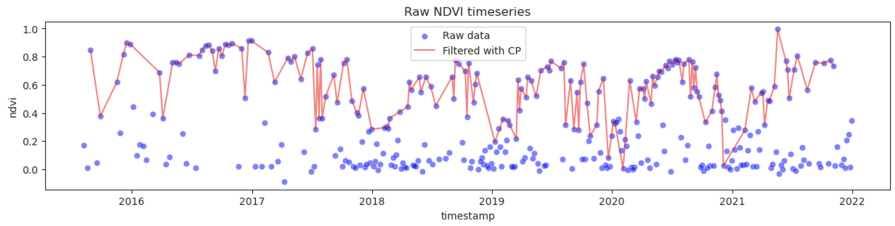



Figure 4.42: Raw NDVI signal for Valais

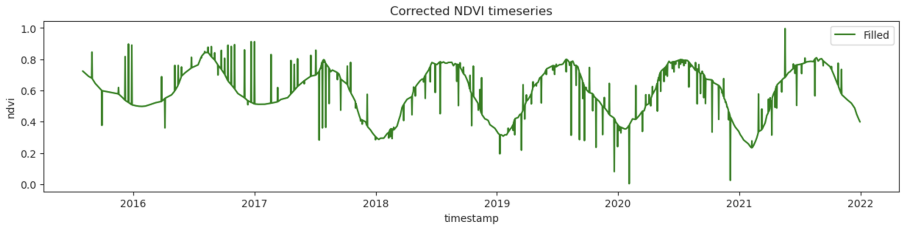



Figure 4.43: Cleaned NDVI for Valais using Savitzky-Golay for smoothing and Linear Interpolation for interpolating

and contain 49% percent of forest. All the data cubes are pre-computed and saved on the hard drive of the cluster. Total number of data cubes created for this project is **eleven thousand**, each having a volume of **70 MB**. In order to be flexible as possible, data cubes that have been created includes all of the features that can be seen in Table 4.3, both for labels and data.If a feature needs to be removed for an analysis, it is done right before feeding the data to the model.

# Chapter 5

# Approach

## 5.1 Guided Prediction

The aim of this research is to use past and future data to predict satellite imagery, specifically the Normalized Difference Vegetation Index from Sentinel 2 using future weather information from ERA 5, then detecting the anomalies in the NDVI that can allow us to predict drought. From the temporal perspective, this problem can be modelled as a spatio-temporal sequence problem in which the task is to predict future n-frames from historical data, also called n-step ahead prediction. There are two approaches to this task: the recursive approach and the one-shot approach. In the recursive approach, pre- dictions for each time step are made sequentially, using the predictions from the previous time steps as input. For example, if the goal is to predict time steps t + 10, t + 20, and t + 30, the prediction for t + 10 is used as input to predict t + 20, and the prediction for t + 20 is used as input to predict t + 30. However, in the one-shot approach, all of the predictions are made at once using the historical data as input. The one-shot approach can handle different context and target data, but it has more parameters to optimize and is therefore more complex. On the other hand, the recursive approach is less complex but may suffer from error accumulation as the horizon gets larger, leading to decreased performance. Which approach to take is heavily dependent on the task and the data. One of the key difficulties in this scenario is to effectively integrate future weather predictions within the framework of the past data and model architecture, while ensuring that there is no contamination of data due to information leakage. An illustration of the task is presented in Figure 5.1.

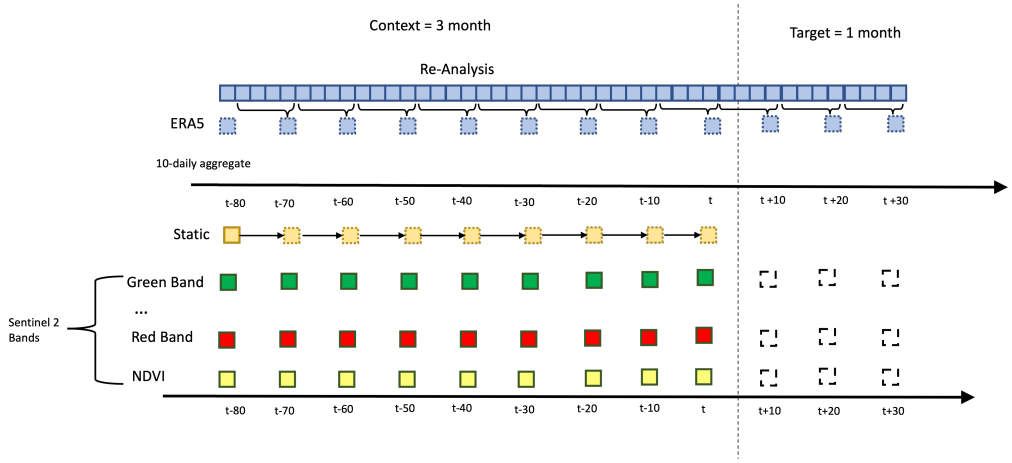



Figure 5.1: To predict the NDVI at times $t+10$, $t+20$, $t+30$ future data from ERA5 that is available at $t+10$,$t+20$, and $t+30$ must be included.

This task has previously been referred to as "Guided Prediction" in the literature, where the model is guided with future information during the prediction process Requena-Mesa et al. (2020). This challenge has also been observed in other domains too, such as video frame prediction where the goal is to predict the future frame while leveraging the edge information from the future Wang et al. (2018). Kladny et al. (2022) has also faced the same challenge in an Earth observation setting. Guided prediction is challenging due to the complex structure of the data, which consists of multiple time series with varying lengths and modalities. Trying to utilize future data in the modelling part creates a mismatch shape of the data, and this creates issues with accurately modeling and predicting the labels. To address this challenge, Kladny et al. (2022) shifted the to-be used future-data to align the time-series of different modalities. Here, we follow the same approach, and *shift the future ERA5* data so that it is aligned with the time-series of the NDVI data. An illustration of the shifting operation can be seen in Figure 5.2.

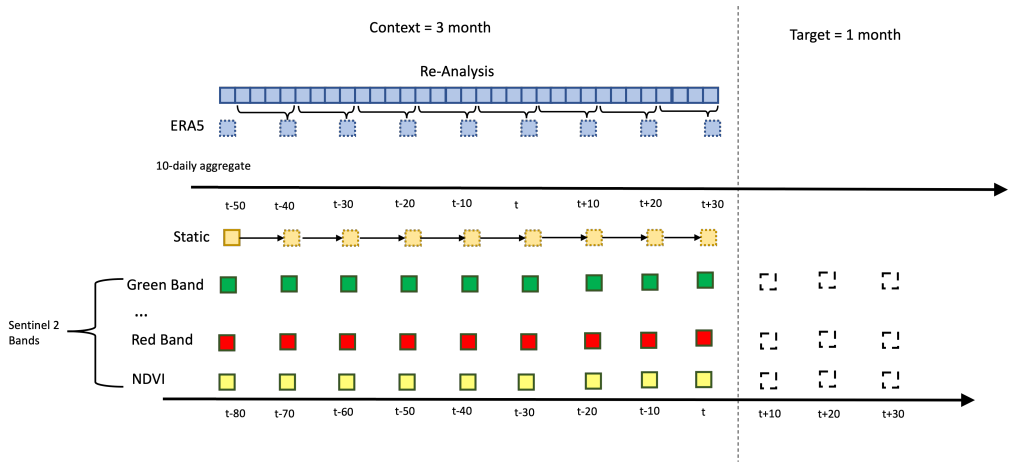



Figure 5.2: ERA5 is shifted to align the timeseries

The use of shifting operations effectively resolves the issue of misaligned time-series. However, this approach also introduces a new challenge in the form of data loss, and causality. As demonstrated in Figure 5.2 as an example, the shift operation results in the loss of historical information from time-steps *t-80* to *t-60*. The amount of data loss is directly proportional to the number of time-steps being predicted in advance. Second issue arises in causality. As mentioned earlier, the goal is to take advantage of the future weather information of the prediction day of interest. Nevertheless, as illustrated in Figure 5.2, predictions of *t+10* are also informed by the weather at *t+20* and *t+30*, not only the weather at *t+10*. This creates a temporal inconsistency, as the NDVI values at time-step *t+10* cannot logically be affected by weather events that occur after *t+10*. This complicates the modeling process and requires careful consideration in the design of the predictive model.

The aforementioned issues arise when attempting to use a one-shot approach to predict the value of a variable for an entire month. The recursive approach also suffers from these issues of time-series alignment and information flow,but just for one-timestamp at a time. However, this approach introduces a new challenge in the form of an imbalanced number of input-output channels. For instance, if one decides to use 26 channels of Sentinel 2 data as input, but only wants to predict one of the variables (e.g, NDVI), a problem arises when attempting to use the predicted NDVI values to make further predictions, as we do not have the future values of the other 25 Sentinel 2 channels.To address this issue, we must either pad the input data with placeholder values to fill the non-predicted channels or predict all of the bands. The latter approach is not feasible due to the large amount of required computational resources and memory, while the former may introduce excess noise into the model. As a result, we must carefully consider which of these two strategies to adopt.

In our model, the initial set-up was to use 9 timestamps (3 months), to forecast 3 timestamps (1 month), with all the features included. However, due to the challenges we have faced with both on the memory capacity and modelling, we decided to simplify the problem, then increase its complexity as we progressed. By simplifying the problem first, we aimed to create a foundation or base upon which we could build and expand. In order to simplify the problem, we limited the number of days to forecast and the number of features to use. The simplified set-up is the following:

- **Number of time steps to forecast:** 1
- **Temporal resolution of the dataset:** 10 days
- **Number of historical time steps to use:** 9
- **Is data cleaned from the clouds? :** Yes
- **Number of features to use:** 7
  - NASA DEM
  - Forest Mask (WSL)
  - Solar Radiation (ERA 5)
  - Surface Pressure (ERA 5)
  - Temperature (ERA 5)
  - Total Precipitation (ERA 5)
  - NDVI (Sentinel 2)

When we limited the model to forecast only one timestamp, then the recursive vs. one-shot approach challenge is simplified, since forecasting one timestamp corresponds to the same task in both of the approaches. However, in the future, as the task becomes more complex, in order to do not suffer from information leakage problem, its chosen to use recursive approach.

## 5.2 Drought Modelling

In this work, we are following Brun et al. (2020)'s methodology as explained in Chapter 3 to identify droughts. We applied a similar analysis as Brun et al. (2020) to our data for the Vaud region, and compared the difference in NDVI between years 2017 and 2018. Before applying the analysis, we masked the data with cloud probability and forest masks to have a cloud-free data set with a significant amount of forests. The result can be seen in Figure 5.3. From the results, in

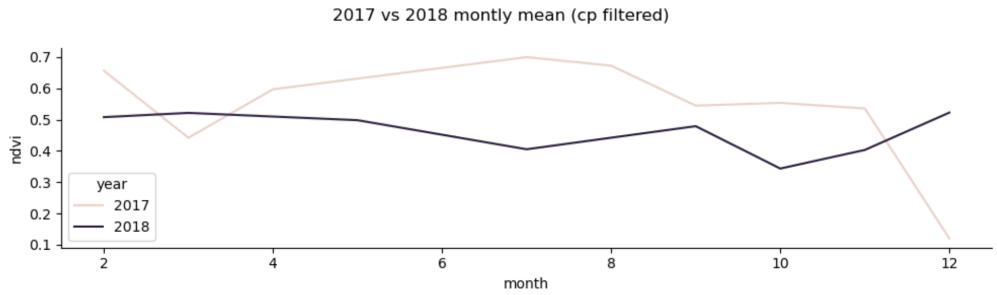



Figure 5.3: Mean NDVI Comparison between 2017 (non-drought year) and 2018 (drought year)

summer months (between June(6) and August (8)) we observe a difference of $\sim 0.2$. Therefore, in our methodology, we would consider a drought event to occur if there is a deviation of 0.2 from the mean NDVI of the non-drought year 2017.

We see drought impact prediction as a two-fold problem. The first step is to predict NDVI, which can be seen as a simple regression problem. From there we move on to identify the impacts based on our NDVI predictions in the way proposed above. As mentioned in Chapter 3, Convolutional LSTM models are a popular choice when it comes to satellite image prediction, and they create the basis of our proposed model. Since we are working with a many-to-one prediction setting, we can formulate satellite image prediction task a sequence modelling problem. Hence, we propose an encoder-decoder Convolutional LSTM architecture (see Figure 5.4). Data are first encoded through Encoder ConvLSTM cells, extracting features using convolutional kernels, and capturing temporal dependencies using LSTM-like gate operations. Then ConvLSTM decoders uses the encoder state(s) as input and procceses these iteratively through the various cells to produce the output. The output of the final decoder ConvLSTM cell then goes in to a 3D CNN, to output desired number of output channels. In order to assess how well the model performs it must be compared to a baseline. To assess our model, we propose a baseline that outputs the last available date in our data. As a general seasonal trend, NDVI values gradually rise from spring to rainy season and fall steadily from autumn to winter Ghebrezgabher et al. (2020). Hence we do not expect to see a drastic change from one day to another. Considering that we have data for nine timestamps timestamps for a single month and we are forecasting the upcoming date with a $\Delta T$ of ten days, in theory there should not be a significant difference in terms of NDVI between $t-1$ (last available historical date) and $t$ (forecasting date). Hence, as a baseline to see how well our model works, we consider a model that outputs the NDVI value at $t-1$,and from there we calculate the loss between the label NDVI and baseline model output, and compare our predictions with the baseline.

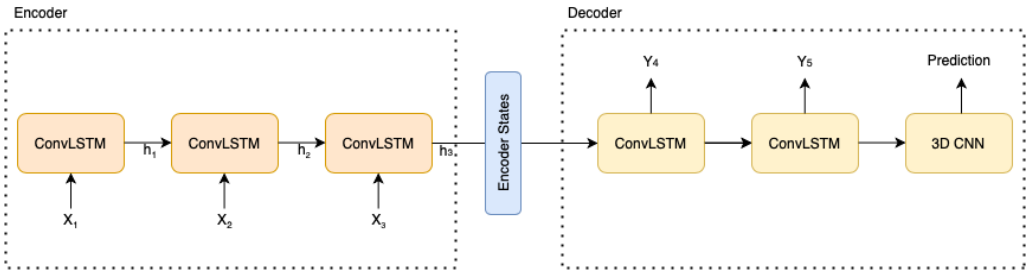



Figure 5.4: Encoder-Decoder ConvLSTM Architecture

# Chapter 6

# Experiments and Results

In this chapter we will explain the experiments that have been conducted, and our results. In all of the experiments, one data sample contains 9 timesteps of historical data, and 1 timestep of forecasting horizon. With a $\Delta T$ of 10, this corresponds to "learning from 3 months of historical data to forecast ten days ahead.". Historical data contains 7 features as mentioned in Chapter 3, and labels only contain NDVI feature. The overall pipeline that contains all the steps until conducting experiments can be seen in Figure 6.1.

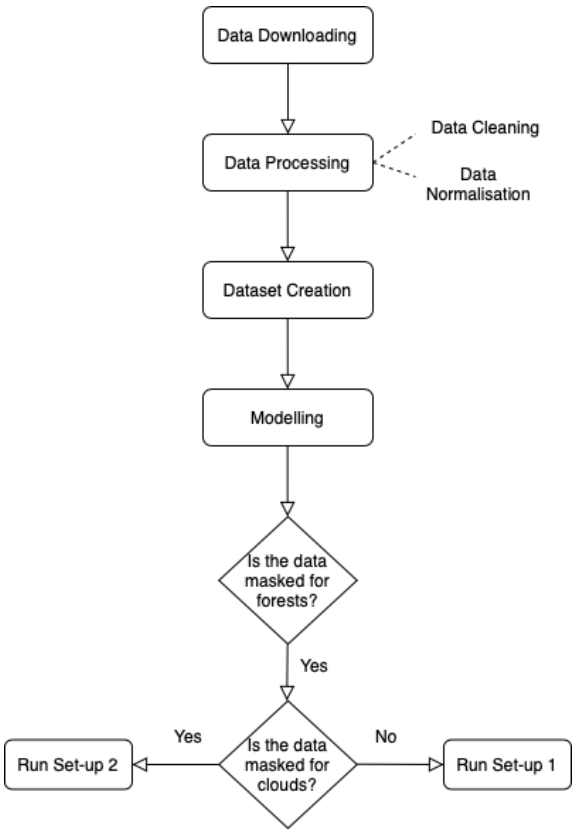



Figure 6.1: Overall pipeline

## 6.1 Experiments

In this experiment we are trying to answer the research question: **How well do deep learning models predict NDVI under sparsity conditions (clouds)?**. Initial plan was to run two models. In one model while calculating the loss only the pixels that are forests will be included, and in the second model only the pixels that are forests, and are cloud-free will be taken into account, and finally the performance of the both models would be compared All the experiments are done by the same Auto-encoder Convolutional-LSTM model. Parameters of the model can be seen in Table 6.1.

| **Hyperparameter** | **Value** |
|---|---|
| Learning Rate | 0.0003 |
| Optimiser | Adam |
| Number of Layers | 3 |
| Kernel Size (ConvLSTM) | (3,3) |
| Kernel Size (3D CNN) | (3,3,1) |
| Hidden Dimensions | 16 |
| Batch size (Train) | 4 |
| Batch size (Validation/Test) | 4 |
| Number of training samples | 7000 |
| Number of validation/test samples | 1000 |
| Number of epochs | 23 |

Table 6.1: Model parameters

All of the experiments are conducted on IBM Research - Cognitive Computing Cluster. The Cognitive Computing Cluster, is a research computing cluster consisting of both x86 and PowerPC compute nodes equipped with Nvidia GPUs. Its primary function is to enable the execution of large-scale and small-scale research experiments, particularly those that require ultra-fast network connectivity and cutting-edge GPU technology. We used NVIDIA V100 TENSOR CORE GPU with eight GPUs, and 256GB of RAM to run our experiments. For each run, model is trained and validated for 23 epochs that in total took approximately **20 hours** to compile.

## 6.2 Results

The original plan was to run two models, as was described. However, the model we were planning to run with both cloud and forest masking, was unsuccessful. In the discussion chapter, we will explore the various factors that may have contributed to the outcome. Hence, we focused on the model that is masked only for forest. Here, we will show our results for the forest-masked model. All the results with the evaluation metrics can be seen in Table 6.2.

For testing, we used a sample from Jura region,2020. The historical dates are: **2020-06-13,2020-06-23,2020-07-03,2020-07-13,2020-07-23,2020-08-02,2020-08-12, 2020-08-22,2020-09-01**, and we are trying to forecast the date **2020-09-11**. In Figure 6.2, we can see the not-imputed NDVI scene (before cleaned for clouds), its corresponding cloud probability mask, and the cleaned ground truth label. For evaluation, we are keeping track of Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and model persistence. We use "persistence" as a baseline to our model. It is the loss which is MSE between the last historical date and our ground truth label as also explained in

Chapter 5.

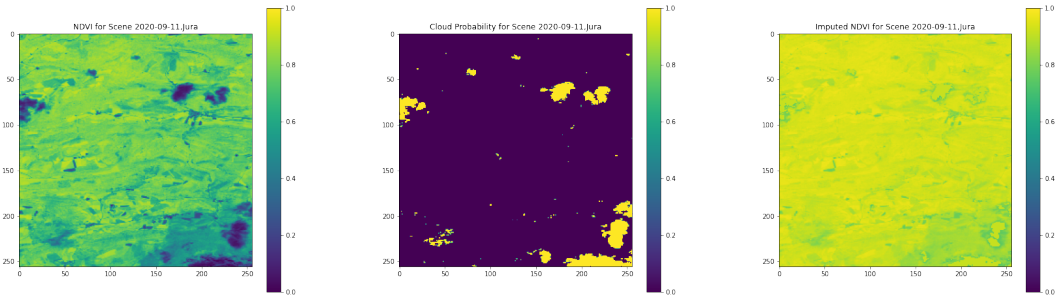



Figure 6.2: Label for 2020-09-11: Original NDVI scene as label (Left), the cloud probability mask used to filter with a threshold of 30% and impute the cleaned signal (Middle), and the resulting corrected NDVI scene as label (Right) in the Jura region on 2020-09-11

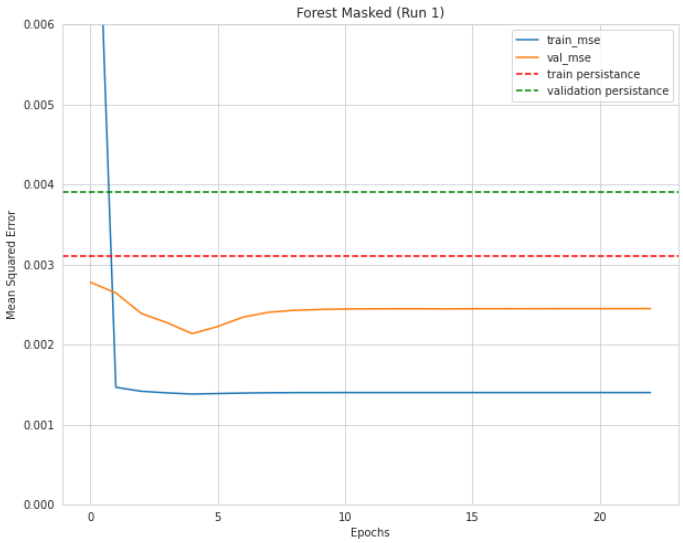



Figure 6.3: MSE and persistence for train and validation

The results for the Mean Squared Error compared with persistence can be seen in Figure 6.3. As we can see, there is a sudden drop in the training performance and an increase in the validation. However, both for training and validation we can see that our model has a superior performance compared to the baseline. Here we are **testing** our model with a **cleaned** data sam-

ple (scene). When we compare the model's prediction and the ground truth label, we see that model's prediction is heavily influenced by the "imputed NDVI" value. (see Figure 6.4). In this test sample our model had a **MSE of 0.0003** while **Persistence is 0.0001**, meaning that our model performed worse than a model that outputs the last day in our data.

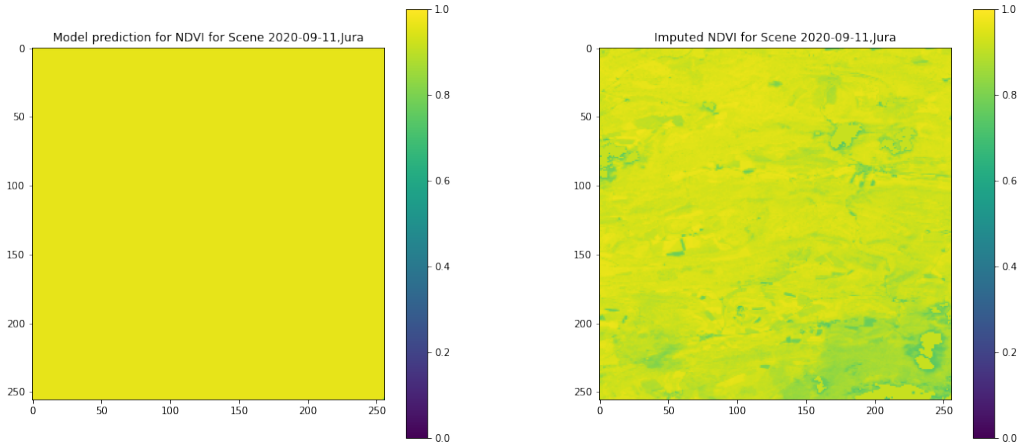



Figure 6.4: Model prediction (Left), and ground truth label (Right).

We also applied a pixel-wise analysis to see how model predicts one NDVI value for a pixel that has high cloud probability in the same predicted sample. Results for the analysis can be seen in Figures 6.5. In the Figure 6.5, we show the evolution of one NDVI pixel in one testing sample, along with the cloud probability. In the Figure 6.5, When we compare the models pixel-wise prediction with the ground truth value, we see that the model almost predicted the exact same value as the ground truth. However, for a pixel in the scene that has a low cloud probability (%30), from the Figure 6.6 we see that models prediction is off from the ground truth.

| **Split** | **Metric** | **Value** |
|---|---|---|
| Train | MSE | 0.001 |
| | RMSE | 0.032 |
| | Persistance (calculated by MSE) | 0.003 |
| Validation | MSE | 0.002 |
| | RMSE | 0.04 |
| | Persistance (calculated by MSE) | 0.004 |
| Test | MSE | 0.004 |
| | RMSE | 0.06 |

Table 6.2: Resulst of the forest-masked model

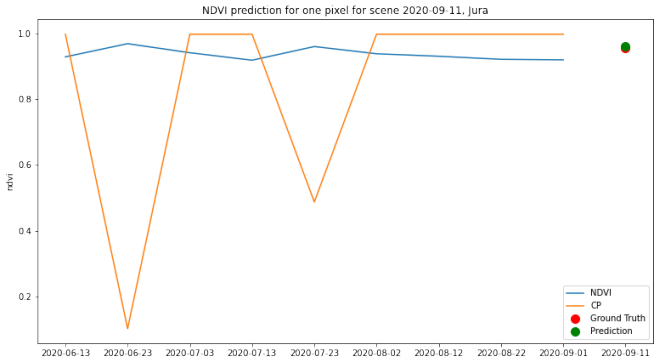



Figure 6.5: Model's prediction for timestamp 2020-09-11 of a pixel with a high cloud probability (90%). Graph also shows the changes in NDVI and cloud probability over time, as represented by the blue and orange lines in a single data sample, respectively.

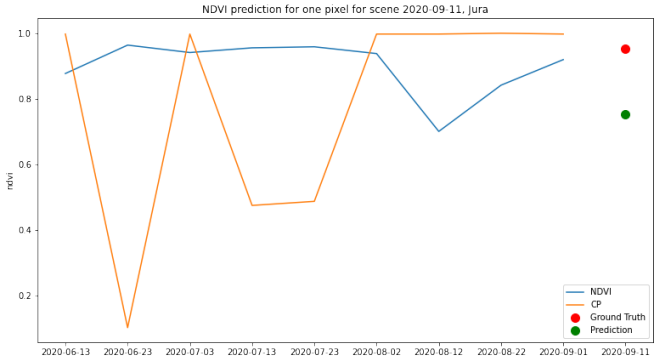



Figure 6.6: Model's prediction for timestamp 2020-09-11 of a pixel with a low cloud probability (30%). Graph also shows the changes in NDVI and cloud probability over time, as represented by the blue and orange lines in a single data sample, respectively.

# Chapter 7

# Discussion

The main objective of our research was to forecast drought impacts in four regions of Switzerland - Jura, Schaffahusen, Vaud, and Valais - using NDVI as a proxy. Our approach to addressing this task was a two-stage method. The initial phase was to forecast NDVI, which can be seen as a regression task, and then detect anomalies to recognize drought impacts. However, at present, our focus is on the first step of the process - predicting NDVI - and we still have a ways to go before we can accurately predict anomalies. Throughout our research, we encountered numerous data and modeling challenges and learned some valuable lessons.

## 7.1 Data Operations

The first step in this work was to download the data, create the data sets, and apply data processing techniques. In the data downloading section, the first challenge we faced was the irregularities in temporal and spatial resolutions (see Chapter 4,Section 4.3). From the spatial aspect, in order to match the resolutions and also because of the memory limitations, we had to coarse-grain the data. This can be seen as a form of data loss since we are not leveraging the high resolution of the data layers. Another approach could be to keep the resolutions at their original levels while handling the difference in the modeling part. We were not able to explore this in this research; however, we do believe that this would be a valuable thing to investigate. From a temporal aspect, we identified irregularities in different swaths in Switzerland (see Chapter 4, Section 4.2). In order to simplify the problem, we chose regions that have similar temporal characteristics, however, this also meant that we could not use all of the data we have.

As mentioned, all the data samples are created beforehand and saved to hard disk. Initially, the goal was to do the data operations on the fly. That is, everything related to the creation of data samples and the loading of data batches would take place while the algorithm was running. However, we discovered during the experiments that this was a time-consuming operation. One epoch took around textbf 24 hours for a simple model with only five features, which was not desirable. We believe that the reason for this is that everything related to the data set, and the data sampler was hand-crafted (see Chapter 4). Thus, they were not optimized in the best possible way, causing a long runtime for the models. Because of these limitations, all the data had to be pre-computed. However, this came with its own challenges. Initially, we intended to create at least **20 thousand** samples for training, and for testing and validation **4 thousand samples** each, with each sample having a volume of **70 MB**. This equals to **2 TB** of memory space just for the data itself. This was not feasible given the infrastructure we had, hence number of data samples had to be limited. We wanted our data samples (scenes) to have a large number of pixels that are forests for experiment purposes. However, our sampling strategy was not optimally implemented; hence, finding a scene that has 80-90% forest was challenging and time-consuming,

| Year | NDVI Anomaly Values | **Drought**-Level |
|---|---|---|
| 2015 | 12.75 | Non-drought |
| 2016 | 9.58 | Non-drought |
| 2017 | -3.77 | Mild drought |
| 2018 | -0.44 | Mild drought |
| 2019 | 3.09 | No drought |
| 2020 | -7.17 | Mild drought |
| 2021 | -14.83 | Moderate drought |

Table 7.1: NDVI anomaly identification according to Nanzad et al. (2019)

so we had to limit our data samples to contain at least **49%** forests. This, of course, affected the quality of our experiments.

Another challenge we faced was statistically and visually "identifying" droughts in our data set. We have tried different methodologies, both statistically and visually. We applied different visualization techniques, including box plots, histograms, line charts, and scatter-plots. However, due to the large amount of noise in the data that comes from clouds and similar atmospheric artifacts, we could not observe any distinct anomalies that could be connected to droughts. We also conducted statistical tests, including the T-test and Mann-Withney U-tet, to see if there is a significant amount of difference between years that could lead to drought events. However, tests showed that all the years were statistically different from each other. Hence, we could not observe droughts statistically either. After these outcomes, we decided to apply methods that have been used in the literature, as explained in Chapter 3. To identify droughts, we applied the steps suggested by Nanzad et al. (2019). We sampled a region in Jura, Switzerland, where clouds and forests are masked using Sentinel 2 and WSL Forest mask data, and applied the same approach. Results can be seen in Table 7.1. Unfortunately, we cannot use information about whether a year was considered a "drough year" in Mongolia by Nanzad et al. (2019) for interpreting our results in Switzerland. The prior knowledge we have from the studies that have been done specifically on Switzerland Brun et al. (2020); Schuldt et al. (2020) do not match the results we obtain from following Nanzad et al. (2019)'s approach. This could be because this study was conducted in Mongolia, and every region has its own characteristics regarding vegetation. It is quite possible that one scheme will not fit all geographic locations. In conclusion, this approach resulted in the failure to identify NDVI anomalies in our data set. When this approach failed, we decided to follow Brun et al. (2020)'s methodology (see Chapter 5,5.2). However, the difficulty we encountered in detecting NDVI anomalies in our data set posed a significant barrier to our progress from NDVI regression to drought impact prediction.

In this work, we are presenting a complete, novel pipeline for generating earth observation data cubes. However, we see that this task itself comes with its own unique challenges. We learned that one of the most important things is scalability when it comes to generating large amounts of data. All of the difficulties we encountered were mostly related to scalability. Of course, data is an essential part of every research project; hence, the problems we faced here and the way we chose to tackle the issues had a great impact on the quality of the experiments and the model results. Even though we propose a solid methodology to generate earth observation data samples, we believe that there is room for improvement.

## 7.2 Modelling

We moved on to the modeling task after creating our data set for the model prediction, with the goal of forecasting NDVI values ten days ahead ($DeltaT = 10$). From our prior knowledge, we chose to go with the Autoencoder-ConvLSTM model Kladny et al. (2022). As previously mentioned, we planned to run two models to see the effect of clouds. One model with only forest masking and one model with both cloud and forest masking. However, due to the difficulties in cloud masking caused by data processing, we were unable to mask the data for clouds. Hence, we continued with the model where we only use forest-mask to mask our data for loss calculation. We are only seeing the effects of complete forest pixels in our loss function. From the results from Chapter 6, we see that the training and validation MSE are lower than their corresponding persistence values, demonstrating that our model has converged. However, when we checked the outputs (prediction) of the model in the test set (see Figure 6.4), we saw that the model was heavily influenced by the imputed values. Even though the values in the predicted scene are not exactly the same, they are almost identical to the imputed value, which is **0.96**. Instead of learning the true signal, our model learned the imputed NDVI value. In the same predicted scene, we also examined the model predictions at the pixel level. We saw that the model's prediction was fairly close to the ground-truth NDVI value when the pixel had a high cloud probability, hence it predicts well when the pixel value is imputed. This supports our hypothesis that instead of the true signal model, it learned the imputed value. We believe there are several reasons for this, and the challenges we have faced in the data operations, as mentioned previously, have had a big impact on the model's behavior.

The first reason is that the model we have implemented is a Recurrent Neural Network (RNN)-based model, which also includes convolution operations and is followed by a 3D CNN layer. Hence, we have a complex that has to optimize around **one hundred thousand** parameters. This type of complex model is data-hungry. It means that a large amount of data is needed to train this type of model. As mentioned in previous chapters, we were only able to generate **seven** thousand samples for training and **one thousand samples** for testing and validation, which we believe is insufficient for the model to learn the true signal and improve its generalizability.

Secondly, one of the most important parts of any machine learning project is the hyper-parameter tuning. There is no fixed recipe for which parameters to use; hence, it is important to customize the parameters according to the task. Unfortunately, due to time constraints, we could not perform a hyperparameter tuning step. Instead, we used Kladny et al. (2022) as a foundation for our model, and fixed the parameters by using Kladny et al. (2022) as a reference. However, the two works have different approaches to the task of predicting drought impacts, as well as different data sets. Hence, it is more than possible that the parameters we are using for our model are not optimal.

Another point is data processing. Data processing is an essential part of every project, and in this work, as mentioned in the previous chapters, all data sets are handcrafted. One thing that could lead our model to learn the imputed value instead of the true signal is the way we handle cloudy pixels. Here, we chose to impute them with a "generated" clean NDVI signal (see Chapter 4). Because our data contained a large number of clouds, many pixels were imputed. Imputing the cloudy pixels might have reduced the noise in the data that comes from the clouds, but it introduced another type of noise that comes from the "artificial" data that is imputed. Hence, it is more than possible that with this type of "data cleaning," the data and hence the model lost the true sight of the NDVI signal. For more solid results, we believe that, a model that is also cloud masked should be run, and the results must be compared.

Even though the results we got from our forest-masked model showed that the model converged, we believe that this sudden drop in the training and these really small loss values should be investigated further. We learned that cloud removal has an essential role in Satellite Image

Prediction tasks and must be handled with care. The results we got from our model showed that the way we handle clouds is not optimal, and more future research is needed in this direction.

# Chapter 8

# Conclusion and Future Work

In this work our aim was to predict drought impacts. We approached this task as a two-fold problem. Goal was to predict the NDVI on a scene-level without any feature-engineering, and from the NDVI prediction predicting the drought impacts. In this research, we were focusing on the data processing and modelling aspects of the drought impact prediction task. We introduced a complete, novel earth observation data cubes creation methodology including data processing techniques (Chapter 4) to maximize the effectiveness of the machine learning models. We also showed that how can we integrate future data from different modalities to our models to enhance the predictive power of machine learning models (Chapter 5). We created our data set with the methodologies and approaches introduced here, and trained our proposed model for NDVI regression task. Our proposed model was fully data-driven, without feature engineering. In order to asses model's performance under cloud conditions we conducted experiments (Chapter 6). Our experiments showed us our model is heavily influenced in its predictions by the value we used to impute the cloudy pixels. Instead of learning the true NDVI signal, proposed model "memorized" the imputed value. Hence, it performed fairly well when the data had clouds and it was imputed.

In conclusion, we proposed methodologies for creating earth observation data cubes, and for applying "guided-prediction". We trained our model for NDVI regression task, and asses its performance under clouds. We saw that drought impact prediction is challenging task (see Chapter 7) both from the modelling and data processing aspect. Thus, we believe that, there is a lot of room for further research.

Our research and experiments showed us the importance of cloud removal in an earth observation data set. We saw how it is effected the model's performance. As for future work, we highly recommend to deep dive to cloud removal topic itself. Future research should consider the potential effects of clouds and how to mitigate these effects more carefully, for example one could explore different methodologies for cloud removal such as deep learning based techniques Chen et al. (2015) and Li et al. (2021). We think that this could improve the model's performance. Future research might also apply different models to the drought prediction task, instead of using Auto-encoder ConvLSTM models. As future work, we highly recommend to further train the model with more training data and investigate it is performance. We believe that the challenges we faced in terms of number of data samples can be resolved in the future-work. As for data creation methodology we have proposed here, we highly recommend to do further research in terms of scalability. As Discussed in Chapter 7, all the challenges we have faced was regarding scalability. Hence, we believe that further research in the scalability and big-data applications is needed to further improve the pipeline.

# Chapter 9

# Appendix

## 9.1 Explanotary Data Analysis and Data Cleaning

Complete results of the analysis conducted to see how different level of cloud probability thresholds effect the NDVI values in different years can be seen in Figures 9.1, 9.2, 9.3.

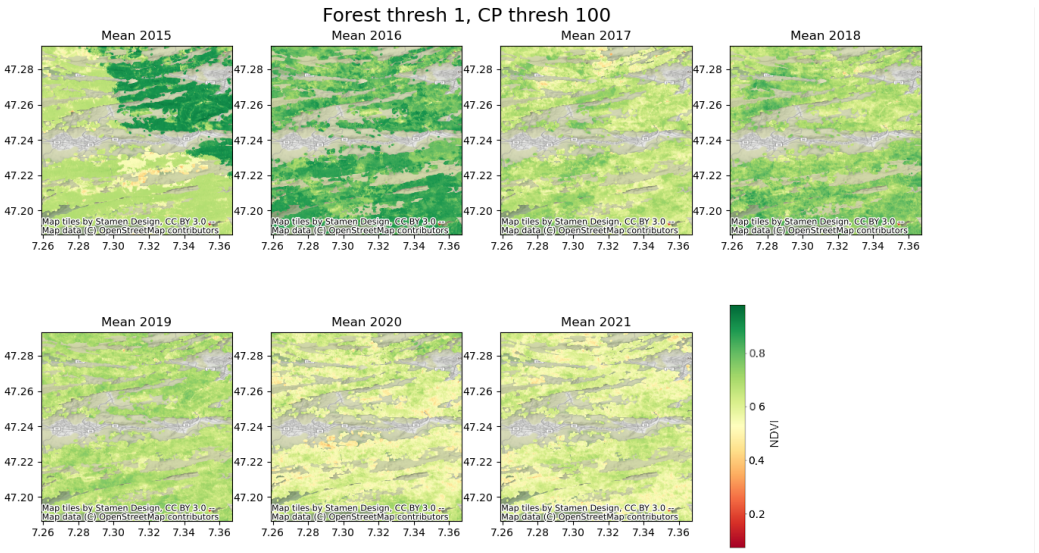



Figure 9.1: Change of NDVI value in one forest pixel over the years with cloud probability threshold 100.

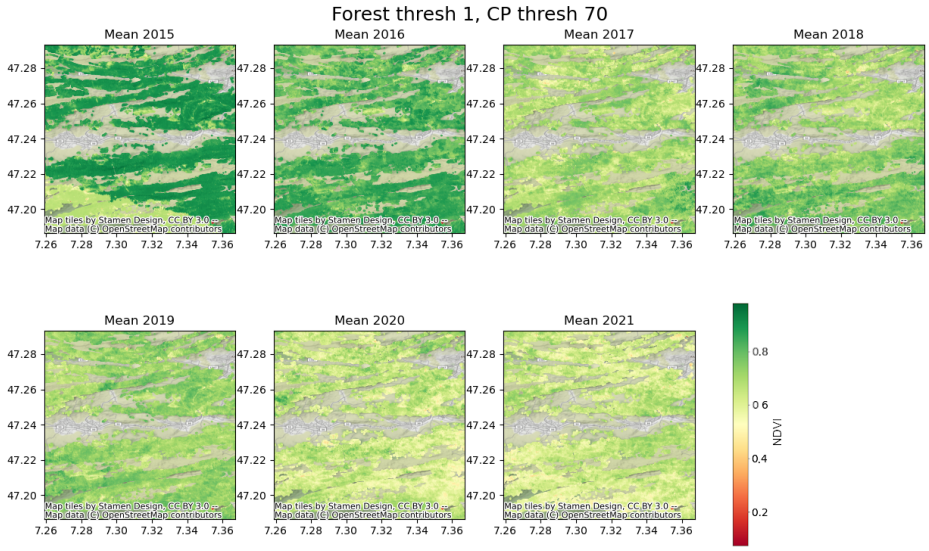



Figure 9.2: Change of NDVI value in one forest pixel over the years with cloud probability threshold 70.

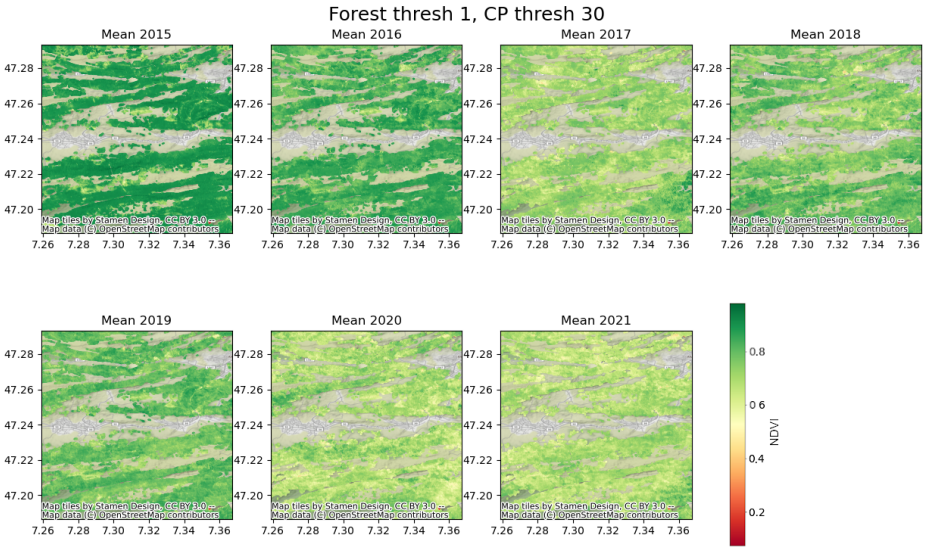



Figure 9.3: Change of NDVI value in one forest pixel over the years with cloud probability threshold 30.

## 9.2 Data Downloading

Complete list of QQ plots and histograms for ERA5 variables can be seen in Figures 9.13, and 9.23

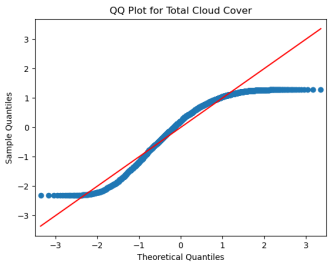



Figure 9.4: QQ Plot for Total cloud cover

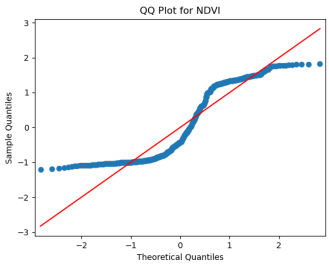



Figure 9.5: QQ Plot for NDVI

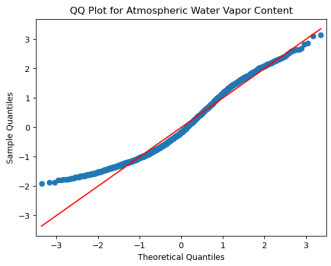



Figure 9.6: QQ Plot for Atmospheric Water Vapor Content

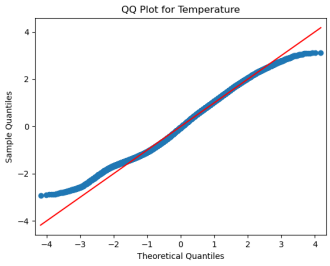



Figure 9.7: QQ Plot for Temperature

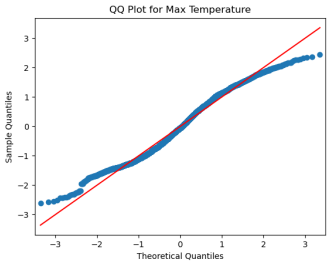



Figure 9.8: QQ Plot for Maximum Temperature

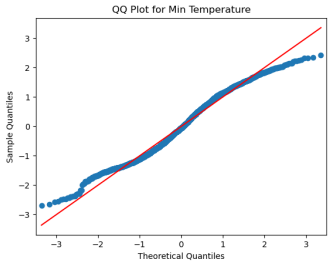



Figure 9.9: QQ Plot for Minimum Temperature Content

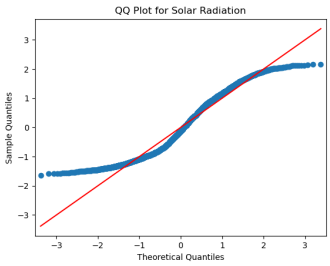



Figure 9.10: QQ Plot for Solar Radiation

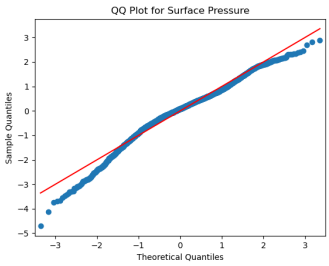



Figure 9.11: QQ Plot for Surface Pressure

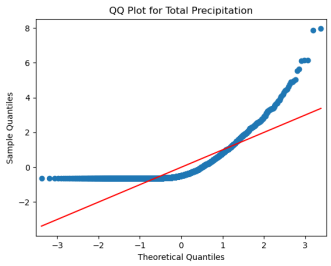



Figure 9.12: QQ Plot for Total Precipitation

Figure 9.13: QQ Plots for ERA5 variables

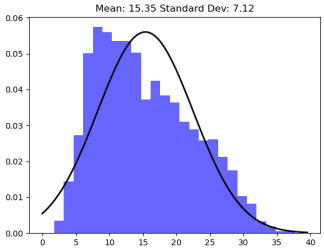



Figure 9.14: Histogram for Atmospheric Water Vapor Content

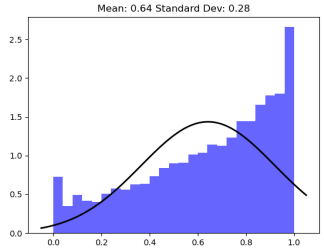



Figure 9.15: Histogram for Total Cloud Cover

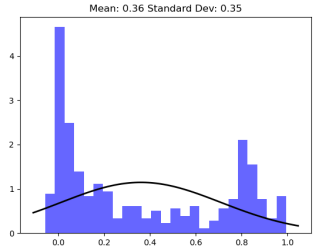



Figure 9.16: Histogram for NDVI

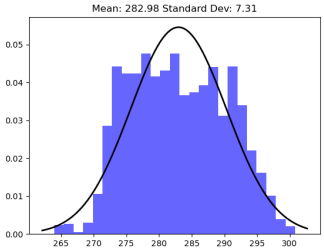



Figure 9.17: Histogram for Temperature

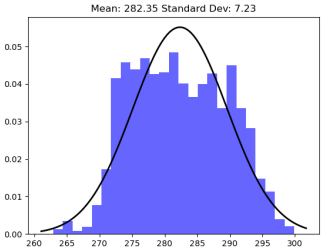



Figure 9.18: Histogram for Minimum Temperature

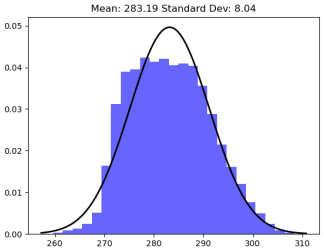



Figure 9.19: Histogram for Temperature Content

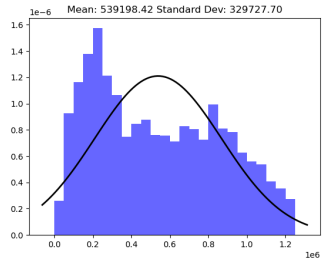



Figure 9.20: Histogram for Solar Radiation

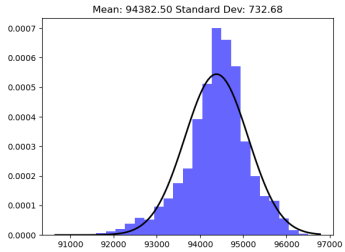



Figure 9.21: Histogram for Surface Pressure

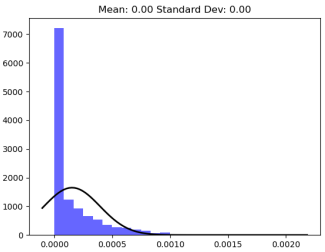



Figure 9.22: Histogram for Total Precipitation

Figure 9.23: Histograms for ERA5 variables

## 9.3 Dataset Creation

Class diagram for the dataset class can be seen in Figure 9.24

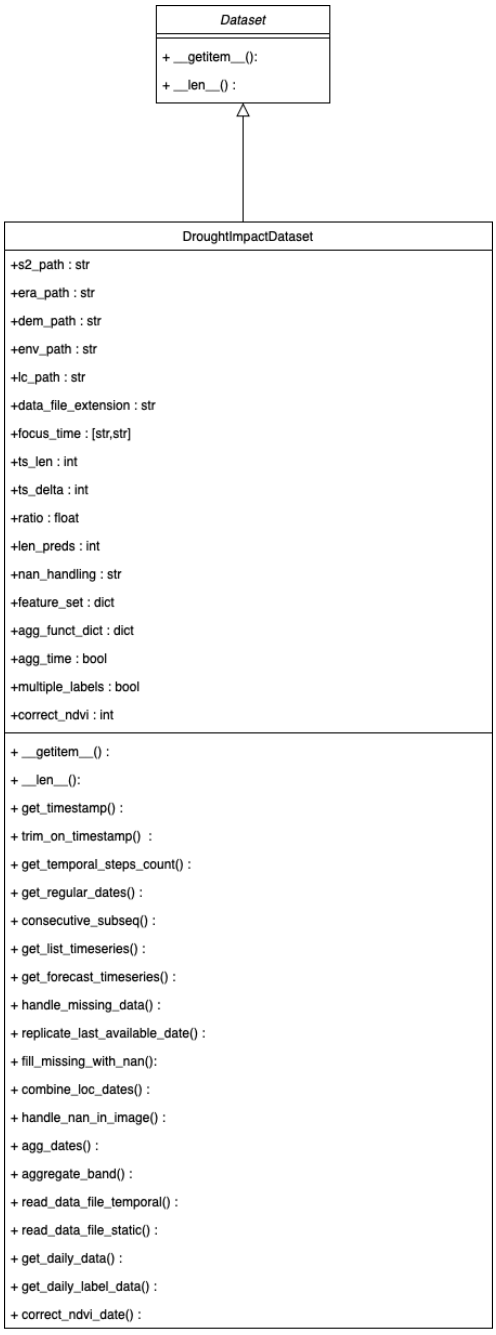



Figure 9.24: Class diagram of dataset class

Class diagram of the data sampler class can be seen in Figure 9.25

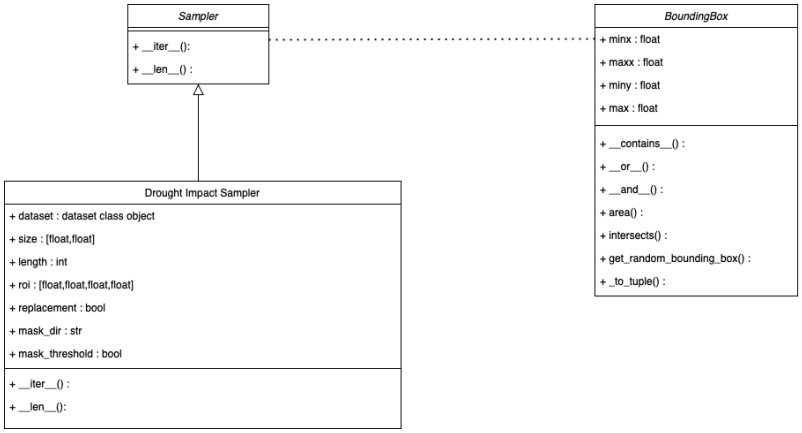



Figure 9.25: Class diagram of data sampler class

# List of Figures

1.1 Multi-dimensional remote sensing data from Ma et al. (2015) . . . 2

3.1 LSTM network structure Col (2015) . . . 10
3.2 An example CNN architecture García-Ordás et al. (2020) . . . 11
3.3 Inner structure of ConvLSTM, taken from Shi et al. (2015) . . . 12
3.4 ConvLSTM Autoencoder . . . 13

4.1 Regions of interest . . . 15
4.2 Sentinel 2 - Normalized Vegetation Index Data Layer for Jura region taken on 08-23-2016 . . . 16
4.3 NASA SRTM Digital Elevation Model - for Jura region . . . 19
4.4 Landcover Classification Map - for Jura region on 2020-08-02 . . . 19
4.5 Temporal inconsistency between years in Sentinel 2 . . . 20
4.6 Sentinel 2 swaths in Switzerland . . . 21
4.7 Time series irregularity between swaths in Switzerland. The yellow circle highlights the temporal inconsistencies between different swaths. . . . 21
4.8 Time evolution of one pixel in Jura region from 2015 to 2022 . . . 22
4.9 Change of NDVI value in one forest pixel . . . 22
4.10 No filtering with CP . . . 23
4.11 CP threshold: 70 . . . 23
4.12 CP threshold: 30 . . . 23
4.13 The effect of different cloud probability thresholds on the NDVI data. Darker shades of green indicate higher NDVI values. . . . 23
4.14 QQ Plot for Total Precipitation . . . 26
4.15 QQ Plot for NDVI . . . 26
4.16 QQ Plot for Solar Radiation . . . 26
4.17 Histograms for some of the ERA5 variables and NDVI Band from Sentinel 2. . . . 26
4.18 Histogram for Total Precipitation . . . 26
4.19 Histogram for Solar Radiation . . . 26
4.20 Histogram for NDVI . . . 26
4.21 Histograms for some of the ERA5 variables and NDVI Band from Sentinel 2 . . . 26
4.22 Mean and Median difference in percentage for Surface Pressure . . . 27
4.23 Mean and Median difference in percentage for Total Cloud Cover . . . 27
4.24 Mean and Median difference in percentage for Temperature . . . 27
4.25 Difference between mean and median for some era variables. . . . 27
4.26 Data Downloading Scheme, configuration 1 . . . 29
4.27 Swath regions 1,2,3 are showing similar behaviour . . . 30
4.28 Swath regions and region of interests . . . 30
4.29 Example downloaded NDVI Layer . . . 31
4.30 Creating data cubes for the region and time of interest (Left) and sampling of smaller cubes from it (Right) . . . 32
4.31 Dataset Class: Step 1. Given the time range and $\Delta T$, dataset class first generates all the possible time series that can be created. . . . 34
4.32 Dataset Class: Step 2 . . . 34
4.33 Data structure produced by dataset class. The four dimensions of the data structure are [time,channels,height,width] . . . 35
4.34 Yellow bounding box is returned by the data sampler . . . 36
4.35 One data sample showing 6 historical (context) dates and 3 forecast (target) dates . 36

4.36 Overall data set creation pipeline . . . 37
4.37 Savitzky-Golay applied signal . . . 38
4.38 Linearly interpolated signal after applying Savitzky-Golay filter . . . 38
4.39 NDVI vs. Cloud Probability thresholds . . . 39
4.40 Mean/Median NDVI vs. Cloud Probability threshold . . . 40
4.41 Data Loss vs. Cloud Probability . . . 40
4.44 Cloudy pixels in the NDVI scene (Left), the cloud probability mask used to filter and impute the cleaned signal (Middle), and the resulting corrected NDVI scene (Right) in Jura region on 2018-04-05 . . . 40
4.42 Raw NDVI signal for Valais . . . 41
4.43 Cleaned NDVI for Valais using Savitzky-Golay for smoothing and Linear Interpolation for interpolating . . . 41

5.1 To predict the NDVI at times $t + 10$, $t + 20$, $t + 30$ future data from ERA5 that is available at $t + 10$,$t + 20$, and $t + 30$ must be included. . . . 44
5.2 ERA5 is shifted to align the timeseries . . . 44
5.3 Mean NDVI Comparison between 2017 (non-drought year) and 2018 (drought year) 46
5.4 Encoder-Decoder ConvLSTM Architecture . . . 47

6.1 Overall pipeline . . . 49
6.2 Label for 2020-09-11: Original NDVI scene as label (Left), the cloud probability mask used to filter with a threshold of 30% and impute the cleaned signal (Middle), and the resulting corrected NDVI scene as label (Right) in the Jura region on 2020-09-11 . . . 51
6.3 MSE and persistence for train and validation . . . 51
6.4 Model prediction (Left), and ground truth label (Right). . . . 52
6.5 Model's prediction for timestamp 2020-09-11 of a pixel with a high cloud probability (90%). Graph also shows the changes in NDVI and cloud probability over time, as represented by the blue and orange lines in a single data sample, respectively. . . . 53
6.6 Model's prediction for timestamp 2020-09-11 of a pixel with a low cloud probability (30%). Graph also shows the changes in NDVI and cloud probability over time, as represented by the blue and orange lines in a single data sample, respectively. . . . 53

9.1 Change of NDVI value in one forest pixel over the years with cloud probability threshold 100. . . . 61
9.2 Change of NDVI value in one forest pixel over the years with cloud probability threshold 70. . . . 62
9.3 Change of NDVI value in one forest pixel over the years with cloud probability threshold 30. . . . 63
9.4 QQ Plot for Total cloud cover . . . 64
9.5 QQ Plot for NDVI . . . 64
9.6 QQ Plot for Atmospheric Water Vapor Content . . . 64
9.7 QQ Plot for Temperature . . . 64
9.8 QQ Plot for Maximum Temperature . . . 64
9.9 QQ Plot for Minimum Temperature Content . . . 64
9.10 QQ Plot for Solar Radiation . . . 64
9.11 QQ Plot for Surface Pressure . . . 64
9.12 QQ Plot for Total Precipitation . . . 64
9.13 QQ Plots for ERA5 variables . . . 64
9.14 Histogram for Atmospheric Water Vapor Content . . . 65
9.15 Histogram for Total Cloud Cover . . . 65

9.16 Histogram for NDVI . . . 65
9.17 Histogram for Temperature . . . 65
9.18 Histogram for Minimum Temperature . . . 65
9.19 Histogram for Temperature Content . . . 65
9.20 Histogram for Solar Radiation . . . 65
9.21 Histogram for Surface Pressure . . . 65
9.22 Histogram for Total Precipitation . . . 65
9.23 Histograms for ERA5 variables . . . 65
9.24 Class diagram of dataset class . . . 66
9.25 Class diagram of data sampler class . . . 67

# List of Tables

3.1 Drought classification scheme according to Nanzad et al. (2019) . . . . . 8

4.1 List of data layers from Sentinel 2 . . . . . 17
4.2 ERA5 Climate Reanalysis Model Data Layers and corresponding spatial resolutions 18
4.3 List of data sources and features. The IBM PAIRS resolution is the interpolated resolution not the original spatial resolution of the bands. . . . . 24
4.4 One Sample Kolmogorov-Smirnov Test for ERA5 variables . . . . . 27
4.5 Features and aggregation methods used for ERA5 variables . . . . . 28
4.6 Number of timestamps . . . . . 31

6.1 Model parameters . . . . . 50
6.2 Resulst of the forest-masked model . . . . . 52

7.1 NDVI anomaly identification according to Nanzad et al. (2019) . . . . . 56

# List of Listings

4.1 Example IBM PAIRS query . . . . . 24

# Bibliography


(2015). Understanding lstm networks.

Abtew, W., Melesse, A., Abtew, W., and Melesse, A. (2013). Vapor pressure calculation methods. *Evaporation and Evapotranspiration: Measurements and Estimations*, pages 53–62.

Albawi, S., Mohammed, T. A., and Al-Zawi, S. (2017). Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee.

Alley, W. M. (1984). The palmer drought severity index: limitations and assumptions. *Journal of Applied Meteorology and Climatology*, 23(7):1100–1109.

Anyamba, A. and Tucker, C. J. (2012). Historical perspective of avhrr ndvi and vegetation drought monitoring. *Remote sensing of drought: innovative monitoring approaches*, 23:20.

Berger, V. W. and Zhou, Y. (2014). Kolmogorov–smirnov test: Overview. *Wiley statsref: Statistics reference online*.

Brun, P., Psomas, A., Ginzler, C., Thuiller, W., Zappa, M., and Zimmermann, N. E. (2020). Large-scale early-wilting response of central european forests to the 2018 extreme drought. *Global change biology*, 26(12):7021–7035.

Chen, B., Wu, Z., Wang, J., Dong, J., Guan, L., Chen, J., Yang, K., and Xie, G. (2015). Spatio-temporal prediction of leaf area index of rubber plantation using hj-1a/1b ccd images and recurrent neural network. *ISPRS Journal of Photogrammetry and Remote Sensing*, 102:148–160.

Chiang, J. L. and Tsai, Y. S. (2013). Reservoir drought prediction using two-stage svm. In *Applied Mechanics and Materials*, volume 284, pages 1473–1477. Trans Tech Publ.

Copernicus, E. (2016). Copernicus land monitoring service.

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., and Bharath, A. A. (2018). Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65.

Das, K. R. and Imon, A. (2016). A brief review of tests for normality. *American Journal of Theoretical and Applied Statistics*, 5(1):5–12.

Das, M. and Ghosh, S. K. (2016). Deep-step: A deep learning approach for spatiotemporal prediction of remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 13(12):1984–1988.

Dracup, J. A., Lee, K. S., and Paulson Jr, E. G. (1980). On the definition of droughts. *Water resources research*, 16(2):297–302.

Drusch, M., Del Bello, U., Carlier, S., Colin, O., Fernandez, V., Gascon, F., Hoersch, B., Isola, C., Laberinti, P., Martimort, P., et al. (2012). Sentinel-2: Esa's optical high-resolution mission for gmes operational services. *Remote sensing of Environment*, 120:25–36.

García-Ordás, M. T., Benítez-Andrades, J. A., García-Rodríguez, I., Benavides, C., and Alaiz-Moretón, H. (2020). Detecting respiratory pathologies using convolutional neural networks and variational autoencoders for unbalancing data. *Sensors*, 20(4):1214.

Ghebrezgabher, M. G., Yang, T., Yang, X., and Sereke, T. E. (2020). Assessment of ndvi variations in responses to climate change in the horn of africa. *The Egyptian Journal of Remote Sensing and Space Science*, 23(3):249–261.

Grossiord, C., Buckley, T. N., Cernusak, L. A., Novick, K. A., Poulter, B., Siegwolf, R. T., Sperry, J. S., and McDowell, N. G. (2020). Plant responses to rising vapor pressure deficit. *New Phytologist*, 226(6):1550–1566.

Hao, Z., Singh, V. P., and Xia, Y. (2018). Seasonal drought prediction: advances, challenges, and future prospects. *Reviews of Geophysics*, 56(1):108–141.

Heim, R. et al. (2000). Drought indices: a review. *Drought: a global assessment*, pages 159–167.

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., et al. (2020). The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049.

Hong, S., Kim, S., Joh, M., and Song, S.-K. (2017). Psique: Next sequence prediction of satellite images using a convolutional sequence-to-sequence network. *arXiv preprint arXiv:1711.10644*.

Huang, L., Mao, F., Zhang, K., and Li, Z. (2022). Spatial-temporal convolutional transformer network for multivariate time series forecasting. *Sensors*, 22(3):841.

Hunt, E. D., Hubbard, K. G., Wilhite, D. A., Arkebauer, T. J., and Dutcher, A. L. (2009). The development and evaluation of a soil moisture index. *International Journal of Climatology: A Journal of the Royal Meteorological Society*, 29(5):747–759.

Jarvis, A., Rubiano Mejía, J. E., Nelson, A., Farrow, A., and Mulligan, M. (2004). Practical use of srtm data in the tropics: Comparisons with digital elevation models generated cartographic data.

Kladny, K.-R. W., Milanta, M., Mraz, O., Hufkens, K., and Stocker, B. D. (2022). Deep learning for satellite image forecasting of vegetation greenness. *bioRxiv*.

Klein, L. J., Marianno, F. J., Albrecht, C. M., Freitag, M., Lu, S., Hinds, N., Shao, X., Bermudez Rodriguez, S., and Hamann, H. F. (2015). Pairs: A scalable geo-spatial data analytics platform. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 1290–1298.

Li, J., Wu, Z., Hu, Z., Li, Z., Wang, Y., and Molinier, M. (2021). Deep learning based thin cloud removal fusing vegetation red edge and short wave infrared spectral information for sentinel-2a imagery. *Remote Sensing*, 13(1):157.

Ma, Y., Wu, H., Wang, L., Huang, B., Ranjan, R., Zomaya, A., and Jie, W. (2015). Remote sensing big data computing: Challenges and opportunities. *Future Generation Computer Systems*, 51:47–60.

McKee, T. B., Doesken, N. J., Kleist, J., et al. (1993). The relationship of drought frequency and duration to time scales. In *Proceedings of the 8th Conference on Applied Climatology*, volume 17, pages 179–183. Boston.

Medsker, L. R. and Jain, L. (2001). Recurrent neural networks. *Design and Applications*, 5:64–67.

Mukherjee, S., Mishra, A., and Trenberth, K. E. (2018). Climate change and drought: a perspective on drought indices. *Current Climate Change Reports*, 4(2):145–163.

Nanzad, L., Zhang, J., Tuvdendorj, B., Nabil, M., Zhang, S., and Bai, Y. (2019). Ndvi anomaly for drought monitoring and its correlation with climate factors over mongolia from 2000 to 2016. *Journal of arid environments*, 164:69–77.

Narasimhan, B. and Srinivasan, R. (2005). Development and evaluation of soil moisture deficit index (smdi) and evapotranspiration deficit index (etdi) for agricultural drought monitoring. *Agricultural and forest meteorology*, 133(1-4):69–88.

O'Shea, K. and Nash, R. (2015a). An introduction to convolutional neural networks. *CoRR*, abs/1511.08458.

O'Shea, K. and Nash, R. (2015b). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

Palmer, W. C. (1968). Keeping track of crop moisture conditions, nationwide: the new crop moisture index.

Park, H., Kim, K., and Lee, D. K. (2019). Prediction of severe drought area based on random forest: Using satellite image and topography data. *Water*, 11(4):705.

Park Williams, A., Allen, C. D., Macalady, A. K., Griffin, D., Woodhouse, C. A., Meko, D. M., Swetnam, T. W., Rauscher, S. A., Seager, R., Grissino-Mayer, H. D., et al. (2013). Temperature as a potent driver of regional forest drought stress and tree mortality. *Nature climate change*, 3(3):292–297.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Peters, A. J., Walter-Shea, E. A., Ji, L., Vina, A., Hayes, M., and Svoboda, M. D. (2002). Drought monitoring with ndvi-based standardized vegetation index. *Photogrammetric engineering and remote sensing*, 68(1):71–75.

Petrou, Z. I. and Tian, Y. (2019). Prediction of sea ice motion with convolutional long short-term memory networks. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9):6865–6876.

Pettorelli, N. (2013). *The normalized difference vegetation index*. Oxford University Press.

Qi, Y., Zhang, Q., Hu, S., Wang, R., Wang, H., Zhang, K., Zhao, H., Ren, S., Yang, Y., Zhao, F., et al. (2022). Effects of high temperature and drought stresses on growth and yield of summer maize during grain filling in north china. *Agriculture*, 12(11):1948.

Rajitha, K., Mohan, M. P., and Varma, M. R. (2015). Effect of cirrus cloud on normalized difference vegetation index (ndvi) and aerosol free vegetation index (afri): A study based on landsat 8 images. In *2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR)*, pages 1–5. IEEE.

Requena-Mesa, C., Benson, V., Denzler, J., Runge, J., and Reichstein, M. (2020). Earthnet2021: A novel large-scale dataset and challenge for forecasting localized climate impacts. *arXiv preprint arXiv:2012.06246*.

Rhee, J. and Carbone, G. J. (2007). A comparison of weekly monitoring methods of the palmer drought index. *Journal of Climate*, 20(24):6033–6044.

Rhee, J., Im, J., Park, S., Forest, R., et al. (2016). Drought forecasting based on machine learning of remote sensing and long-range forecast data. *APEC Climate Center, Republic of Korea*.

Robin, C., Requena-Mesa, C., Benson, V., Alonso, L., Poehls, J., Carvalhais, N., and Reichstein, M. (2022). Learning to forecast vegetation greenness at fine resolution over africa with convlstms. *arXiv preprint arXiv:2210.13648*.

Rolnick, D., Donti, P. L., Kaack, L. H., Kochanski, K., Lacoste, A., Sankaran, K., Ross, A. S., Milojevic-Dupont, N., Jaques, N., Waldman-Brown, A., et al. (2022). Tackling climate change with machine learning. *ACM Computing Surveys (CSUR)*, 55(2):1–96.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Schuldt, B., Buras, A., Arend, M., Vitasse, Y., Beierkuhnlein, C., Damm, A., Gharun, M., Grams, T. E., Hauck, M., Hajek, P., et al. (2020). A first assessment of the impact of the extreme 2018 summer drought on central european forests. *Basic and Applied Ecology*, 45:86–103.

Sheffield, J., Goteti, G., Wen, F., and Wood, E. F. (2004). A simulated soil moisture based drought analysis for the united states. *Journal of Geophysical Research: Atmospheres*, 109(D24).

Sherstinsky, A. (2018). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *CoRR*, abs/1808.03314.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c. (2015). Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28.

Sruthi, S. and Aslam, M. M. (2015). Agricultural drought analysis using the ndvi and land surface temperature data; a case study of raichur district. *Aquatic Procedia*, 4:1258–1264.

Stewart, A. J., Robinson, C., Corley, I. A., Ortiz, A., Ferres, J. M. L., and Banerjee, A. (2022). Torchgeo: deep learning with geospatial data. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, pages 1–12.

Sturm, J., Santos, M. J., Schmid, B., and Damm, A. (2022). Satellite data reveal differential responses of swiss forests to unprecedented 2018 drought. *Global Change Biology*, 28(9):2956–2978.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Liu, G., Tao, A., Kautz, J., and Catanzaro, B. (2018). Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*.

Xiao, C., Chen, N., Hu, C., Wang, K., Xu, Z., Cai, Y., Xu, L., Chen, Z., and Gong, J. (2019). A spatiotemporal deep learning model for sea surface temperature field prediction using time-series satellite data. *Environmental Modelling & Software*, 120:104502.

Xu, Z., Du, J., Wang, J., Jiang, C., and Ren, Y. (2019). Satellite image prediction relying on gan and lstm neural networks. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE.

Zhang, R., Chen, Z.-Y., Xu, L.-J., and Ou, C.-Q. (2019). Meteorological drought forecasting based on a statistical model with machine learning techniques in shaanxi province, china. *Science of The Total Environment*, 665:338–346.