TURUN YLIOPISTO

Agricultural event detection from the satellite image zonal statistics

Mikael Roto

Master's thesis
May 2023

Examiners:
Associate professor Henri Nyberg
Professor Ion Petre

THE DEPARTMENT OF MATHEMATICS AND STATISTICS

UNIVERSITY OF TURKU
Department of Mathematics and Statistics

MIKAEL ROTO: Agricultural event detection from the satellite image zonal statistics
Master's thesis, 38 pages, including 4 pages of attachments.
Mathematics
May 2023

Satellite images have become an important tool for event detection and monitoring. The key advantage for the satellite based monitoring is their ability to cover large areas frequently which in turn makes them very cost-efficient solution for monitoring geographically large areas. Due to advances in the satellite technology and the image processing techniques, satellites are capable of providing high resolution data within real-time from the Earth's surface.

In this thesis we provide a brief introduction to the satellite based remote sensing and how these methods can be used to model different agricultural events. We inspect theoretical satellite signal responses to a common agricultural events and try to detect these patterns from our own dataset.

We develop a method to process satellite images into signals and apply preprocessing methods to increase signal to noise ratio. We then train a gradient boosting classifier to the smoothened signals and process the individual predictions so that we can detect the start and end times for various agricultural events from the agricultural parcels.

Keywords: Remote sensing, machine learning, time series, classification.

Contents

1 Introduction 1

2 Data collection 2
2.1 Signal terminology - Sentinel 1 2
2.2 Signal terminology - Sentinel 2 3
2.3 Signal extraction 5
2.4 Ground truth dataset 5

3 Literature review 7
3.1 Signal behaviour around the events 7
3.2 Modelling attempts 8

4 Methods 10
4.1 Gradient boosting 10
4.2 Evaluation metrics 12
4.3 Savitzky-Golay filter 12

5 Analysis 13
5.1 Raw datasets 13
5.2 Relative orbit numbers 14
5.3 Signal responses to events 15
5.4 Feature generation 16
5.5 Creating the train and test splits 20
5.6 Predictive model 21
5.7 Hyperparameter optimization 23
5.8 Fitting the model 24
5.9 Comparing aggregated predictions 26

6 Discussion 28
6.1 Improvements 28
6.2 Conclusions 30

References 32

Appendix: Feature importances 35

1 Introduction

Common Agriculture Policy (CAP) is a program in the European Union aiming to improve European agricultural productivity, competitiveness and sustainability by range of measures including direct payment, market measures and rural development. Majority of the 59 billion euro budget is managed and controlled by Integrated Administration and Constol System (IACS) whose function is to support farmers to submit their declarations and safeguard CAP financials.

The legal framework of CAP was changed in 2020 in order to simplify and modernize CAP. One of the main points of the reform is to increase the role of satellite Earth Observation (EO) for making the IACS more cost efficient. Sentinels for Common Agriculture Policy (SEN4CAP) project started in 2017 and, its main objectives are delivering EO products, services and algorithms to increase efficiency and traceability of IACS. [1]

Sentinel programme is a series of next-generation Earth observation satellites developed by European Space Agency (ESA). The goal of the Sentinel missions is to provide different kinds of observations from Earth. Each one of the sentinel-missions focus on different aspect of Earth observation such as Athmospheric, Oceanic and Land Monitoring. In this study we are mainly interested in Sentinel-1 (S1) and Sentinel-2 (S2).

Sentinel-1 is the first Copernicus Programme satellite launched by ESA. Originally the mission was composed of two satellites, Sentinel-1A and Sentinel-1B, but Sentinel-1B has been retired and currently Sentinel-1A is the only satellite in this mission. Sentinel-1C and Sentinel-1D are in development with plans of launching Sentinel-1C as soon as possible. S1 satellites carry synthethic-aperture radar instrument which is capable of collecting data regardless of the weather and time of day. Spatial resolution of these satellites are down to 5 meters and can cover up to 400 kilometers in width. The orbit has 12-day cycle and completes 175 orbits per cycle. Data collected by S1 satellites has many purposes such as forest, agriculture and water monitoring, emergency response support in event of environmental disasters and climate change monitoring.[2]

Sentinel-2 is a constellation of two identical satellites in the same orbit that collect high resolution, multi-spectral images from the land and coastal areas. The main applications include agriculture, ecosystem monitoring, forest managements and disaster mapping. Using the twin satellites the revisit frequency is 5 days in the majority of land locations with same viewing conditions, but real revisit frequency might be higher due to multiple tracks. [3]

The data collected by Sentinel missions is made easily accessible by policies made by ESA and European Commission and it be can used for scientific, public or commercial purposes for free.[2]

The purpose of this thesis is to analyze data generated by the Sentinel-1 and Sentinel-2 missions and ground data collected from agricultural parcels. At first we do a literature review of proposed solutions to classify various agricultural events from the signals and the compare different methods in classifying the ground status.

This thesis is done in collaboration with Finnish Food Safety Authority in order to assist Finland to comply with the European Common Aggricultural Policy. The methods and analyses presented in this thesis will be tested during the summer of 2023 in Finland.

2 Data collection

2.1 Signal terminology - Sentinel 1

Sentinel-1 missions collect radar data using synthetic-aperture radar (SAR) instrument. SAR instrument can be compared to how bats use echolocation to navigate. Bats create noise that bounces away from walls and reflection can be observed. The same principle applies to SAR but instead of sending noise the satellite sends microwave pulses that bounce back to the satellite. Frequency of the microwave pulse is 5.405GHz meaning that the wavelength of the pulse is around 5.6cm which bypasses clouds effortlessly. The sensor in SAR creates an image from the signals which get reflected back to device. [4]

Different types of materials and surfaces reflect the microwave pulse differently. This is called backscatter signal (BS). For example smooth surfaces such as roads or roofs scatter the signal primarily to other directions and the satellite receives little to no signal back. More rough areas scatter the signal so the satellite receives some echo back. Third type of reflection is double-bounce scattering which occurs when the signal bounces between one or more flat surfaces back to satellite. Double-bounce scattering can be mostly seen in urban areas [5].

Another type of signal product received from Sentinel-1 mission is coherence data. Interferometric coherence measures similarity of two signals taken at different time [6]. For example now S1 produces 12-day coherence signals (COH12) as the Sentinel-1A revisits the same location every 12 days. The 6-day coherence product was available until the Sentinel-1B's SAR instrument broke and the Sentinel-1B was retired [7].

The microwave pulse can be transmitted and received in different polarizations. Polarization refers to the orientation of the electric field of a radar wave as it travels through space. Different polarization combinations respond differently to various materials and can be used to gain more information from the object of interest. For example vertical transmit, horizontal receive (VH) polarization indicate how large proportion of the signal transmitted in vertical polarization got received in horizontal polarization [5].

Different polarization types:

- HH – for horizontal transmit and horizontal receive
- VV – for vertical transmit and vertical receive
- HV – for horizontal transmit and vertical receive
- VH – for vertical transmit and horizontal receive

Sentinel-1 measures VV and VH polarizations over the Europe. Other observation modes are used over the North and the South poles [8]. VV backscattering is most common with rough surface scattering such as bare ground or water and VH

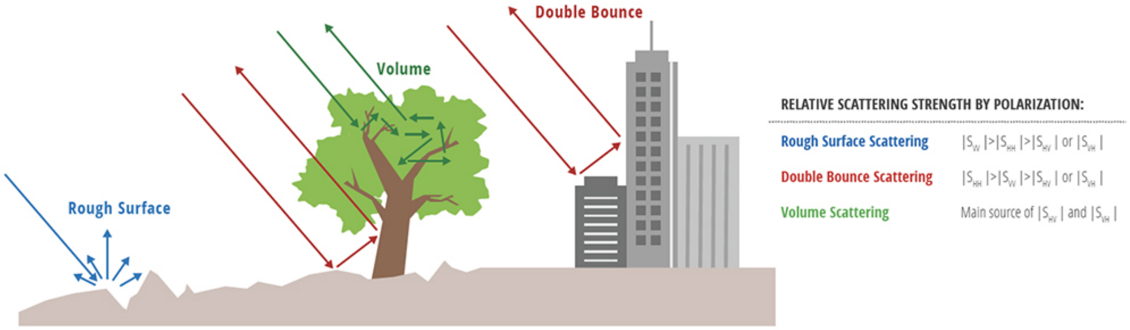
Double Bounce
Volume
Rough Surface
RELATIVE SCATTERING STRENGTH BY POLARIZATION:
Rough Surface Scattering $|S_{VV}|>|S_{HH}|>|S_{HV}|$ or $|S_{VH}|$
Double Bounce Scattering $|S_{HH}|>|S_{VV}|>|S_{HV}|$ or $|S_{VH}|$
Volume Scattering Main source of $|S_{HV}|$ and $|S_{VH}|$

Figure 1: SAR polarization visualised (NASA [5]).

backscattering is more common with more complex materials such as trees or high penetration soil types [5].

These signals are collected over various relative orbit numbers. Relative orbit is a path that S1 satellite passes over an area. When two images are taken from the same relative orbit they have the same incidence angle and look direction which means the images can be compared using multi-temporal analysis. Relative orbit number is calculated from absolute orbit number and this number tells how many times the satellite has passed over this track since its launch. Observations from different relative orbits differ to some degree as the orbits are in different angles. Also the satellite passes each one of the relative orbits from ascending or descending direction which may have an effect on the signal [9]. For example, in Figure 2 the roof of a barn can be seen inside the parcel boundaries as the satellite comes in from a certain angle but is not necessarily seen from all of the orbits.

2.2 Signal terminology - Sentinel 2

Introduction to Sentinel-2 data presented in this Section is based on technical guide to Sentinel-2 by ESA [3]. Sentinel-2 missions collect data using MultiSpectral Instrument (MSI). The MSI instrument measures Earth's reflected radiance in 13 different spectral bands. Spatial resolution of the image ranges from 60 meters down to 10 meters and is higher for some bands and smaller for others. These different bands and resolutions for Sentinel-2A are documented in Table 2. Spatial resolutions are the same for the Sentinel-2B but there is marginal differences in measured central wavelenghts.

In principle the MSI-instrument works similar to a camera but measures more wavelenghts than just standard red, green and blue. You can assemble a RGB image from the S2 images using bands four, three and two which correspond to red, green and blue channels of electromagnetic spectrum.

From these bands we can create different indices that should measure different things in ground. For example Normalized Difference Vegetation Index (NDVI) ranges from $[-1, 1]$ and measures the amount green vegetation in the area. The indices are not the main subject of this study, but those indices that are used in this study and what they measure are included in Table 1.

Index	Measures
BSI	Bare soil index
BSI RGB	Bare soil from index RGB
CIRE	Chlorophyll and vegetation
CRC	Crop residue cover
NDTI	Cultivated land
NDVI	Normalized difference vegetation index
NDVI-RE3	Vegetation, uses different bands compared to NDVI
NSSI	Non-photosynthesizing vegetation

Table 1: Indices created from the S2 bands. The indices and associated references are documented in [10].

Band number	Central wavelength (nm)	Spatial resolution (m)
1	442.7	60
2	492.7	10
3	559.8	10
4	664.6	10
5	704.1	20
6	740.5	20
7	782.8	20
8	832.8	10
8a	864.7	20
9	945.1	60
10	1373.5	60
11	1613.7	20
12	2202.4	20

Table 2: Sentinel-2A bands, central wavelengths and spatial resolutions.

For example the NDVI should decrease and BSI should rise after the mowing event as there is less vegetation and more bare soil. Similarly the CIRE should be lower in the autumn when compared to summer as there is less green vegetation in the fields.

2.3 Signal extraction

The data is received in a set of 2-dimensional arrays where each value correspond to a square in a real world area (e.g. 10m by 10m). Each value in these arrays corresponds to single measurement such as 12 day interferometric synthetic-aperture-radar coherence (COH12) or backscatter. These arrays are received periodically so we can form a time series from these signals which can be used to infer changes in the landscape [11].

In this format the data is too big to store and process efficiently so there is a need to compress this to a lower dimension. In our approach signals are aggregated inside the parcel and descriptive statistics such as minimum, maximum, standard deviation, median, quantiles and quartiles are collected. In Figure 2 there are five parcel outlines plotted on top of the VH backscatter image. For each parcel, we collect the pixel values that land inside the geometry and compute descriptive statistics. These statistics are referred to as zonal statistics.

One key observation from Figure 2 is that aggregated values might be quite noisy. If we take a look at the lowest parcel in the image we see that the parcel outline intersects with object that has quite high backscatter. That object turns out to be a roof of a barn that barely intersects with the parcel from a certain satellite angle and reflects the pulse back at the satellite. In the analysis we try to combat these kinds of outliers by using median of the signals instead of mean.

The S2 signals come from different images and products but the general idea for signal extraction is the same for all of the signals. Importantly S2 indices are first calculated per pixel basis and after that the indices are extracted from the images.

2.4 Ground truth dataset

The ground truth dataset is collected by the Centers for Economic Development, Transport and the Environment (ELY centers) during the summer of 2022. The purpose of the dataset is for development, improvement and validation of land area monitoring algorithms. Every week during the data collection period a human data-gatherer drove a premeditated path and took a record about the state of land cover in each parcel along the route. If the land cover looked the same from week to week the data-gatherer recorded the same event. For example if the parcel looked like it had been mowed for three weeks straight we have data that the parcel was moved for three weeks straight.

General areas where ground truth data was collected is visualized in Figure 3. We can see that most of the data was collected in the southern parts of Finland. The primary reason for this is that the data collection routes begin from locations close to more populated areas and the paths are designed so we can collect as much as possible data under a specied time limit. This approach for the data collection

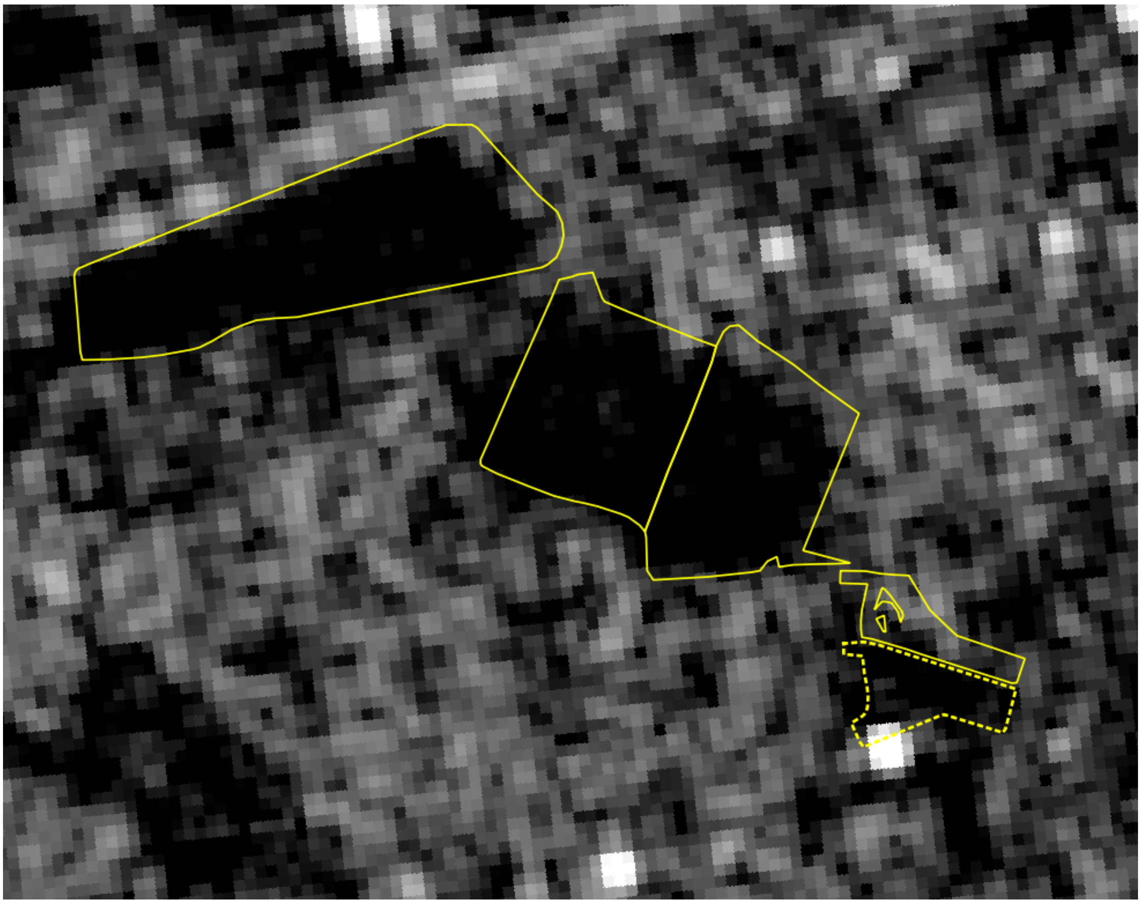

Figure 2: Parcel outlines on top of VH backscatter image. Parcel outlines are in yellow, high backscatter values in white and low values in black.

may introduce certain biases to the dataset, as the areas which are not located near major highways or population centers are underpresented. In this thesis we assume that all agricultural events look the same in across Finland although it may be beneficial to conduct further research in order to investigate the homogeneity of events and improve the model by taking regional differences to account.

The ground truth data collection was a success for the most of it. Due to a large data-gathering there are some discrepancies between the data collected. Some ELYs were more accurate than others and some did not have as regular data collection periods as others. There are also some differences inside the individual ELYs as the data-gathering was not done by a single individual. For example the question “The parcel has been recently mowed?” is quite subjective as there is no clear definition what is recently mowed and there is no accurate information on mowing time without questioning the landowner.

From the weekly collection of the events to parcels we interpolate the events so that we have one event for each day. This might create some data points which have wrong label and we need to take this into account in our analysis. For example, let’s say there is a weekly visit record the events of the parcel every monday. If the grass looks like it is growing on Monday but it is cut next Thursday the status persists

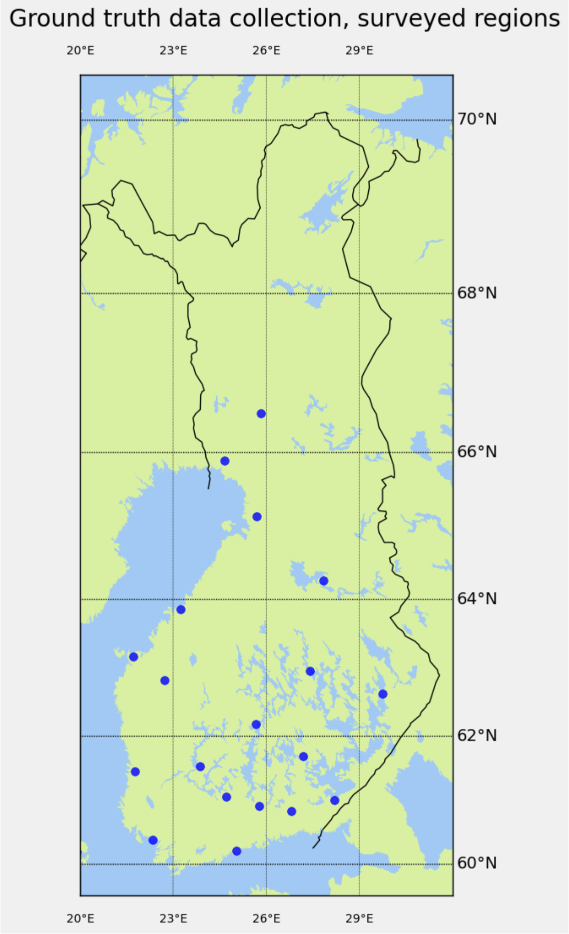

Figure 3: Locations where the ground truth data were collected.

until next Monday.

The grass parcel events were collected from late June to late autumn. The data collection for cereals started from ripening (typically around end of July or beginning of August) and were observed weekly until the end of October. In total there are around 6900 indidivual agricultural events logged from 2400 agricultural parcels. Mean number of visits for a parcel during the data collection period is 14 times and this typically means we have around 100 days worth of data from the status of parcel. Most of the data was collected during mid to late summer which means we have no ground truth data on events from early summer. The ground truth dataset is not available to the public and it is property of the Finnish Food Safety Authority.

3 Literature review

3.1 Signal behaviour around the events

According to Voormansik et al. [12] the coherence-signal should drop before the ploughing event and rise sharply after it. The authors mention that the coherence dropping before the ploughing event may be attributed to other management practices taking place near the ploughing event, such as seeding or cultivation. It is also noted in the study that the VH-polarization should respond to a ploughing event more strongly compared to a VV-polarization. This differs from the signal response with respect to a mowing event where both polarizations respond similarly.

In the case of a mowing event Voormansik et al. found out that coherence is stable until the mowing event and after that rises, but not as sharply as with the ploughing event. Both polarizations respond similarly to the mowing event and the study found no difference between the different polarizations in any point in the signal. Do note that Voormansik et al. used 6 day coherence data but due to retirement of Sentinel-1B in this study we have access only to a 12 day coherence.

Coherence measurements taken from different relative orbit numbers should behave similarly but not in sync as the measurements are taken at different time. One can think that the different orbits measure some latent change in the ground that is masked by noisy measurements. These measurements may also be out of sync so one potential idea for finding the events from the coherence data is to combine these different measurements to cut out noise in individual orbits and predict the event from the combined signals.

Voormansik et al. also state that daily precipitation affects coherence and subsequently rainfall just before the Sentinel-1 data capture can hide the farming event. The rainfall provides noise that need to be taken account when preprocessing the signals. The farming event may be recognisable from the other relative orbit numbers as the article found out that the effect of rainfall to a coherence-signal was stronger in some relative orbits.

3.2 Modelling attempts

There are various methods attempting to predict and model agricultural events from the satellite signal time series. Lobert et al. [13] compared different sets of signals acquired from Sentinel-1, Sentinel-2 and Landsat 8 by using them on one dimensional convolutional neural network [14] and comparing which set of signals receive the best accuracy. The study was conducted on 64 meadows for an overall of 257 mowing events between 2017 and 2019 in Germany.

Major conclusions from the Lobert et al. is that neither optical/SAR alone (Sentinel-1 or Sentinel-2) is not enough alone to classify mowing events. The NDVI is a good input feature to detect mowing events and performs generally better than SAR alone, but underperformed in comparison to optical/SAR combinations.

Lobert et al. used various metrics for evaluating time series predictions such as mean error (ME), mean absolute error (MAE) and normalized mean absolute error (nMAE):

$$
\begin{aligned}
\mathrm{ME} &= \frac{1}{n}\sum_{i=1}^{n}\left(\widehat{Y_i} - Y_i\right), \\
\mathrm{MAE} &= \frac{1}{n}\sum_{i=1}^{n}\left|\widehat{Y_i} - Y_i\right|, \\
\mathrm{nMAE} &= \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\widehat{Y_i} - Y_i}{Y_i}\right|,
\end{aligned}
\tag{1}
$$

where n is the number of meadows or parcels, $\widehat{Y_i}$ is the predicted mowing frequency for each meadow and Y_i is the true mowing frequency. ME is a measure of the average difference between the predicted values from a model and the true values. It is computed by taking the mean of all the differences between the predicted and true values. ME does not give information about the magnitude of the errors, but it does indicate if the model tends to overestimate or underestimate the target values on average. [13]

MAE measures average error regardless of the sign and gives the error in the same units as the prediction. The frequency of the events varies between parcels different parcels so nMAE is useful metric to compare errors between parcels with high and low counts of events. For example if the model is worse at predicting parcels with high number of events the ME and MAE weight these errors highly but with nMAE the error is capped at the maximum of 1 per parcel regardless of event count.[13]

Lobert et al. used only a single relative orbit number for each area from the S1 signals. The relative orbit was chosen so that the orbit covers all of the parcels in the given area and used only ascending orbits as they are generally acquired in the late afternoon in the area of study. The reasoning behind using late afternoons is that the study tried to combat the varying amount of morning dew that might have an effect on the S1 signals.

Lobert et al. also tested Savitzky-Golay filter to smooth all optical and SAR features. Savitzky-Golay is a filter commonly used in signal processing to smooth noisy data. In this context the filter was used to smoothen out noise made by cloudy images or spikes on S1-signals caused by precipitation. Lobert et al. used five for polyorder parameter and seven for filter length parameter. We come back Savitzky-Golay filter more in depth in Section 4.3.

The analysis in Lobert et al. is not directly comparable to our study. In their study they had well-defined date when the mowing events happened compared to our study in which we only had a a time period when the event had happened. But the metrics defined in Equation (1) are directly comparable to our study as they are computed for time periods instead of specific dates. The metrics found in their study are included in Table 3.

Reference mowing frequency	ME	MAE	nMAE
1	0.310	0.369	0.369
2	0.083	0.321	0.161
3	−0.304	0.420	0.140
4	−1.440	1.440	0.360

Table 3: Event detection metrics for best model in Lobert et. al. [13]

4 Methods

4.1 Gradient boosting

This Section presents an introduction to gradient boosting method based approach in Chen et. al. [15]. Given data set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$ $(|\mathcal{D}| = n, \mathbf{x}_i \in \mathbb{R}^m, y_i \in \mathbb{R})$ tree ensemble methods create K additive functions to predict the dependent variable y:

$$\hat{y}_i = \phi(\mathbf{x}_i) = \sum_{k=1}^{K} f_k(\mathbf{x}_i), \quad f_k \in \mathcal{F},$$

where $\mathcal{F}$ is the space of regression trees (also known as CART). The space of regresssion trees is defined by

$$\mathcal{F} = \left\{f(\mathbf{x}) = w_{q(\mathbf{x})}\right\} \left(q : \mathbb{R}^m \to T, w \in \mathbb{R}^T\right),$$

where q denotes the structure of each tree mapping example to the corresponding leaf index and T is the number of leaves in the tree. Each function f_k corresponds to individual tree structure q and leaf weights w. Each regression tree contains a continuous score on each individual leaf and we use w_i to represent score on i-th leaf. For example datapoint in each tree we follow decision rules given by q to classify it into leaves and get the final predictions by summing up the corresponding leaves.

To find the set of functions for this model we have to minimize

$$\begin{gathered}\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k), \\ \text{where } \Omega(f_k) = \gamma T + \frac{1}{2}\lambda \|w_k\|^2 \quad \text{for } k \in \{1, \ldots, K\}.\end{gathered} \tag{2}$$

Here l is differentiable and convex loss function that measures the error between the predictions $\hat{y}$ and the true target values y. The second term is used to penalize the complexity of the model and the complexity of individual trees, which in turn helps the model to avoid overfitting and learning to generalize better. When the regularization parameters γ and λ are set to zero the objective is the same as traditional gradient tree boosting.

The loss function defined in Equation (2) can not be optimized using traditional optimization methods as the parameters include functions. Instead of trying to optimize the function we can build the model in additive manner. Let $\hat{y}_i^{(t)}$ be the prediction for the i-th data point and at the t-th iteration we need to add function f_t to minimize the objective function:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t\left(\mathbf{x}_i\right)\right) + \Omega\left(f_t\right).$$

This leads to approach where we greedily add new decision trees f_t which most improve our model according the loss function defined in Equation (2). We can use second-order approximation to optimize the objective in the general setting:

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^{n} \left[l\left(y_i, \hat{y}^{(t-1)}\right) + g_i f_t\left(\mathbf{x}_i\right) + \frac{1}{2} h_i f_t^2\left(\mathbf{x}_i\right)\right] + \Omega\left(f_t\right), \tag{3}$$

where $g_i = \partial_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right)$ and $h_i = \partial^2_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right)$ are first and second order gradient statistics on the loss function. After removing the constant terms from Equation (3) we get the following objective function at step t:

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n} \left[g_i f_t\left(\mathbf{x}_i\right) + \frac{1}{2} h_i f_t^2\left(\mathbf{x}_i\right)\right] + \Omega\left(f_t\right). \tag{4}$$

If we define $I_j = \{i \mid q\left(\mathbf{x}_i\right) = j\}$ as the instance set of leaf j we can rewrite Equation (4) by expanding Ω term as

$$\begin{aligned}\tilde{\mathcal{L}}^{(t)} &= \sum_{i=1}^{n} \left[g_i f_t\left(\mathbf{x}_i\right) + \frac{1}{2} h_i f_t^2\left(\mathbf{x}_i\right)\right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \\ &= \sum_{j=1}^{T} \left[\left(\sum_{i \in I_j} g_i\right) w_j + \frac{1}{2}\left(\sum_{i \in I_j} h_i + \lambda\right) w_j^2 \right] + \gamma T.\end{aligned}$$

For a fixed tree-structure $q(\mathbf{x})$, optimal leaf weight w_j^* can be computed for j-th leaf by

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda},$$

and the corresponding optimal value is given by

$$\tilde{\mathcal{L}}^{(t)}(q) = -\frac{1}{2} \sum_{j=1}^{T} \frac{\left(\sum_{i \in I_j} g_i\right)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T.$$

By using this function as a scoring function we can measure the quality of tree structure q. The score can be compared to impurity score for decision tree evaluation with the exception that it is derived for a wider range of objective functions. Usually it is impossible to enumerate through all the possible tree structures q and instead we opt to build the branches greedily. The algorithm starts from a single leaf and iteratively adds branches to the tree. This is called greedy strategy as we are making locally optimal choices at each step without considering the entire search space. If I_L and I_R are the sets of left and right nodes after the split and lettting $I = I_L \cup I_R$, then the loss reduction for the split is given by

$$\mathcal{L}_{\text{split}} = \frac{1}{2}\left[\frac{\left(\sum_{i \in I_L} g_i\right)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{\left(\sum_{i \in I_R} g_i\right)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{\left(\sum_{i \in I} g_i\right)^2}{\sum_{i \in I} h_i + \lambda}\right] - \gamma.$$

This formula can be applied to find the best split candidates.

The tree-ensemble methods often achieve higher accuracy than a single decision tree at the cost of interpretability. Following the decision path for hundreds or thousands of decision trees gets out of hand pretty fast.

4.2 Evaluation metrics

Classification errors can be measured by accuracy which is a percentage of correct predictions out of all predictions. This is an intuitive metric but it loses it's meaning quickly if the different classes are unbalanced. For example if we have 100 datapoints where 98 datapoints belong to one class and the remaining two to other. It is quite easy to build a classifier with accuracy of 98% by just classifying all the datapoints to majority class.

In this analysis we will use precision, recall and F1-score. Precision is the proportion of true positives out of all positive predictions. Recall is defined as the proportion of true positives among all the actual positive instances and it measures how well the model identifies the actual positives. F1-score is a way to combine both of these measures into a one metric. By using the harmonic mean the F1-score can not be high without both of the values being high [16]. Equations for the precision, recall and the F1-score are found in Equation (5).

$$\begin{aligned}\text{Precision} &= \frac{tp}{tp + fp},\\ \text{Recall} &= \frac{tp}{tp + fn},\\ \text{F1-score} &= 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}},\end{aligned} \tag{5}$$

where tp indicates the true positives, fp the false positives and fn the false negatives.

4.3 Savitzky-Golay filter

Savitzky-Golay filter is a one-dimensional filter, although it has been generalized for two/threedimensional signal, which can be used to smooth time series data. The primary purpose for smoothing the data is to increase the precision of the data and remove noise. In our example this filter is used to smoothen the noise in the input signals. For example the radar images are quite sensitive to precipitation and optical signals are affected by cloudy images. There is a process where we try to discard cloudy pixels using a cloudmask but some clouds make it through the mask and the filter reduces the effect of these clouds.

The Savitzky-Golay filter achieves data smoothing by fitting a polynomial to datapoint and some surrounding datapoints by method of linear least squares. The order of the polynomial fit and the amount of surrounding datapoints are parameters for the filter. If the datapoints are equally spaced an analytical solution can be found and the filter becomes much faster to calculcate. For evenly spaced datapoints the coefficients for the moving window can be precalculated into a table and they can be used in a convolution.

Convolution is defined as

$$x_j^* = \frac{\sum_{i=-m}^{i=m} C_i \times x_{j+i}}{N},$$

where the index j represents the running index of the ordinate data in the original data table, $C_i, i \in \{1, \ldots, N\}$ are the convolution coefficients and N is the length of the convolution. For example the centered moving average is a convolutional filter where the weights are $C_i = \frac{1}{N}$ and N is the length of the mowing window. The centered mowing average is equal to the Savitzky-Golay filter with the order of the polynomial equal to 1. [17]

5 Analysis

5.1 Raw datasets

At this point our dataset is split to three parts: S1 signals, S2 signals and the ground truth events. The S1 dataset is described in Table 4 and it consist of different radar measurements from various orbits for each one of the parcels. The S2 dataset is more straightforward compared to the S1 dataset. The S2 dataset consists of observation times, parcels and various optical signal index measurements which are described in the Table 1.

The ground truth dataset has been preprocessed to a format where we have parcel id, date and status. For every parcel we have one status for every day throughout the entire data collection period. The raw, i.e. unprocessed dataset is made of weekly data collections and the observations are interpolated so we have one status for each day. The preprocessing is not part of this thesis although improvements to this step could improve results dramatically as the errors in this step are propagated to later parts of the analysis.

Zonal statistic median	Relative orbit	Signal type	Parcel id	Observation time
0.52	130	BS_VH	520	2022-06-23 13:24
0.42	120	BS_VV	520	2022-06-24 01:30
0.1	130	C12_VH	520	2022-06-23 13:24
...	...	...	...	...
0.9	90	BS_VV	130	2022-08-23 03:24

Table 4: The S1 dataset before doing any data manipulation. BS denotes backscatter and C12 is the 12 day coherence.

The major challenge is to normalize the datasets in such way all of the information can be used together. In high level the approach is as follows:

1. Combine the relative orbits together for S1 signals so we have only one measurement from any given day for each signal type.
    - This means that dataset can be transformed into the same shape as the S2 dataset.

- Some orbits are more sensitive to many variables such as precipitation. If we join the measurements from multiple orbits together we reduce random noise. The resulting signal should be more robust to minor events in parcel.
- This step is explored more thorougly in the next Section.

2. Interpolate both S1 and S2 data so we have one value for every day.
   - We used linear interpolation but in further experimentation other interpolation methods could be explored.
3. Apply a Savitzky-Golay filter for all of the signal types.
   - This does not change the format of the data and only the values are changed.
4. Finally join the ground truth status to this dataset by merging the two aforementioned datasets to ground truth by parcel id and date.
   - The resulting dataset is described in the Table 5.
   - In each row we have S1 and S2 signal values and the corresponding date, parcel and event information.

5.2 Relative orbit numbers

In Figures 4 and 5 there are coherence and backscatter signal values plotted from various different relative orbits. The two different mowing events happened between (2022-06-28 to 2022-07-08) and (2022-07-27 to 2022-08-03). On the first mowing event there is a sharp rise in both backscatter values but it is hard to distinguish the other mowing event from the noisy signals.

The approach in this thesis is to combine the information from different relative orbits together and use the combined signals as an input in our model. The way we approach combining the orbits into one signal is by linearly interpolating them into one signal and then smoothing out the output. Each one of the signals is quite noisy on their own but when we combine them together we find major changes in the signals and the minor deviations smoothen out. With the optical signals from S2 there is no need to combine any of the signals, but smoothing can be helpful in processing the signals. The optical signals suffer greatly from clouds and we try to filter these out in the signal extraction process but the cloud mask is not perfect. By applying smoothing into optical signals we can reduce the effect of clouds and find the responses to real changes in the ground.

We can see from Figure 6 that there is some notable peaks in the backscatter signals. The parcel in Figures 6 and 5 is a grassland parcel that had cutting events between (2022-06-28 to 2022-07-08) and (2022-07-27 to 2022-08-03) although there might have been more events before 21.6. and some after September as we have no ground truth data from these time periods. We see that the preprocessed signal peaks after the cutting events but this is not as clear if we look at raw signals in Figure 5.

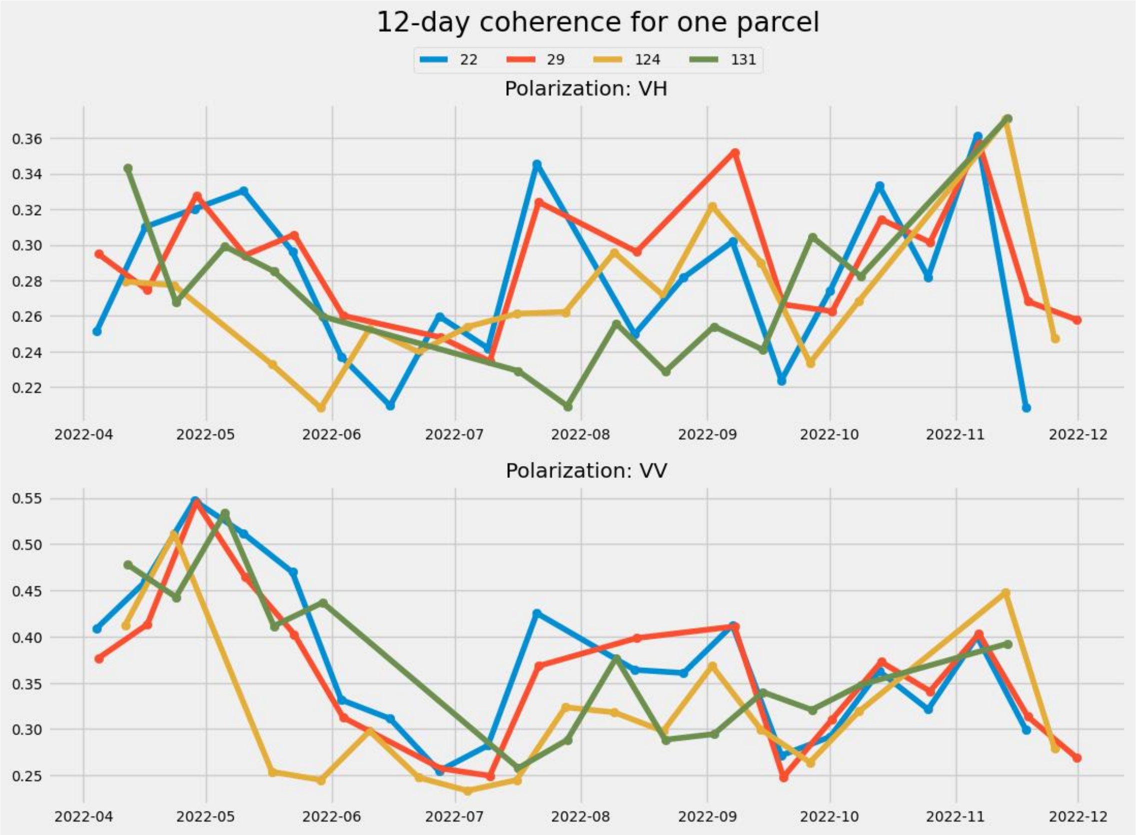

Figure 4: Twelve day coherenses plotted for individual orbit numbers which overlap the parcel. Different colors correspond to different relative orbits. Parcel contains two mowing events which happened between (2022-06-28 to 2022-07-08) and (2022-07-27 to 2022-08-03).

5.3 Signal responses to events

The effects on S1 signals described by Voormansik et al. are also observed on our dataset. In Figure 7 we notice that 12 day coherence rises sharply after the ploughing event. The VV-polarization responds more to the the ploughing compared to VH-polarization. VV-backscatter is noticeably lower before the event and raises after the event. VH-backscatter does not seem to have response to ploughing event. Note that the signal responses in figures of this section are aggregated across all of the relative orbits.

Signal responses to mowing are not as coherent as they are with ploughing event. We can notice from the non-normalized Figure 8 that the signal has larger variance at the time of the event compared to ploughing event. This may be attributed to the fact that most of the mowings are on grassland parcels and the observations from grassland parcels are more heterogenic compared for example to cereals. Although the signals are quite noisy there is some evidence that coherence rises after the mowing event. In Figure 8 we see that quantiles of normalized C12 signal are above zero after 20 days meaning that the coherence should rise.

As coherence measures similarity between two signals we should expect the co-

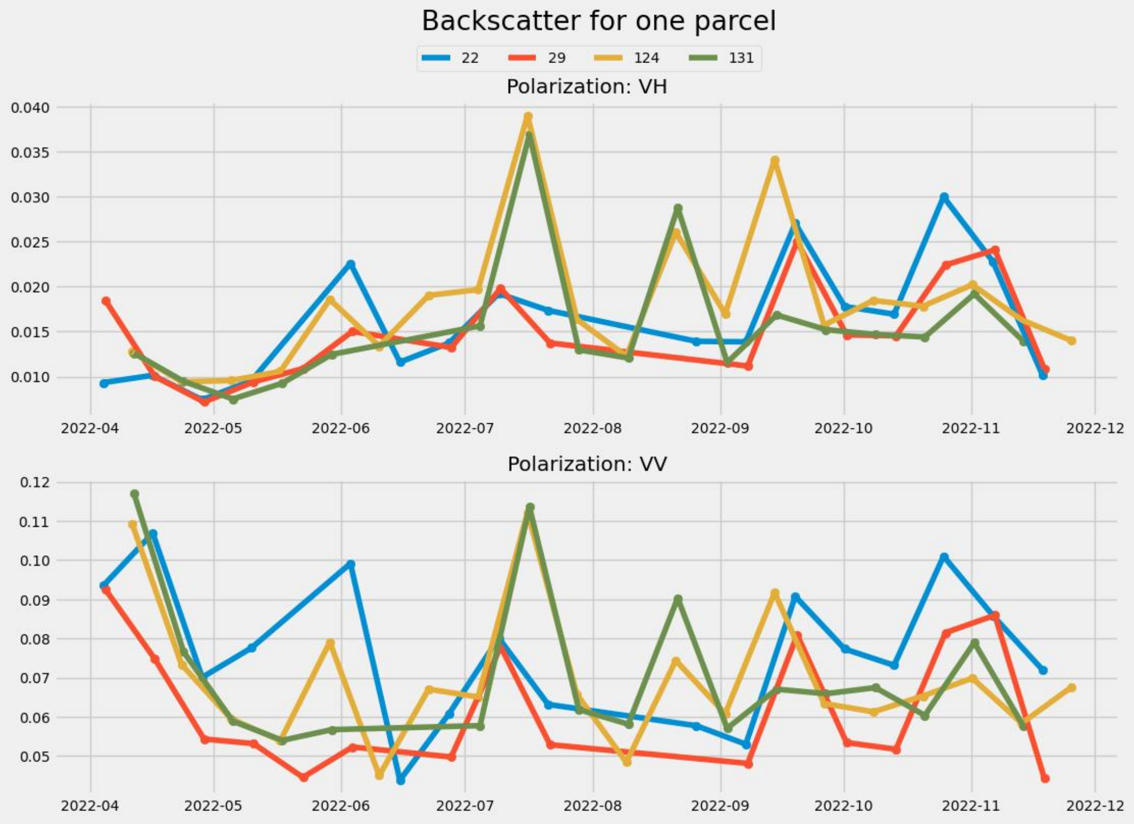

Figure 5: S1 signal backscatters plotted for all relative orbits which overlap the parcel. Different colors correspond to different relative orbits. Parcel contains two mowing events which happened between (2022-06-28 to 2022-07-08) and (2022-07-27 to 2022-08-03).

herence to momentarily decrease and then start rising. Right after the event the coherence should be lower compared to a baseline as it is calculated between grassland and mowed or ploughed ground. Some time after the event the coherence should rise as it is calculated between two mowed/ploughed parcels. One explanation why the grassland does not have such clear response to event is that the grass is left to ground after the mowing so the coherence does not change as much or the grass is collected on some parcels and left on the ground for some parcels and consequently we cannot detect this from the aggregated data.

5.4 Feature generation

At this point our dataset looks like the example in Table 5. Date and parcel id columns are not used as a features in our model so the feature matrix at this point consists of S1 signals and indices created from the S2 signals. Now we are going to create more features out of the existing features.

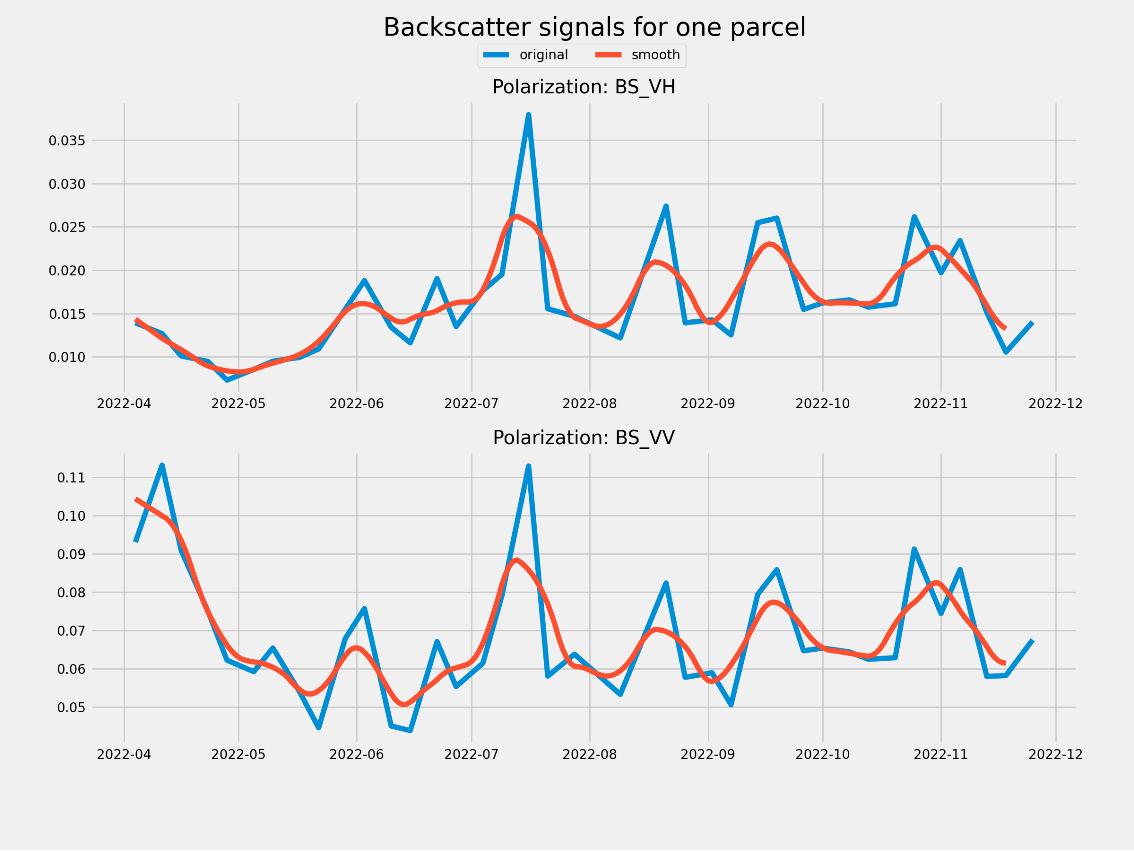

Figure 6: Signal preprocessing for the signals shown in Figure 5. In blue the signals are interpolated together and in red there is a smoothing function applied to interpolated values.

Date	Parcel id	BS_VV	BS_VH	NDVI	...
30.6.2022	521	0.2	0.5	0.7	...
31.6.2022	521	0.21	0.49	0.71	...
⋮	⋮	⋮	⋮	⋮	...
31.10.2022	20520	0.7	0.6	0.1	...

Table 5: The dataset after interpolation and smoothing, but before the feature generation.

In the previous two sections we looked at the signal time series and concluded that observing how signal changes is a valid way to determine if there was some agricultural event on the parcel. The absolute signal values are useful features but it also might be useful to observe the changes in signals. To add temporal dimensional information to our model we can calculate how the signal changes between two days and add it as a feature to our model. For example we can calculate how much signal value has changed from 10 days ago compared to today by taking difference of signal value from 10 days ago and today. If the change is positive it means that the signal value has risen which might indicate some kind of agricultural event. To make this

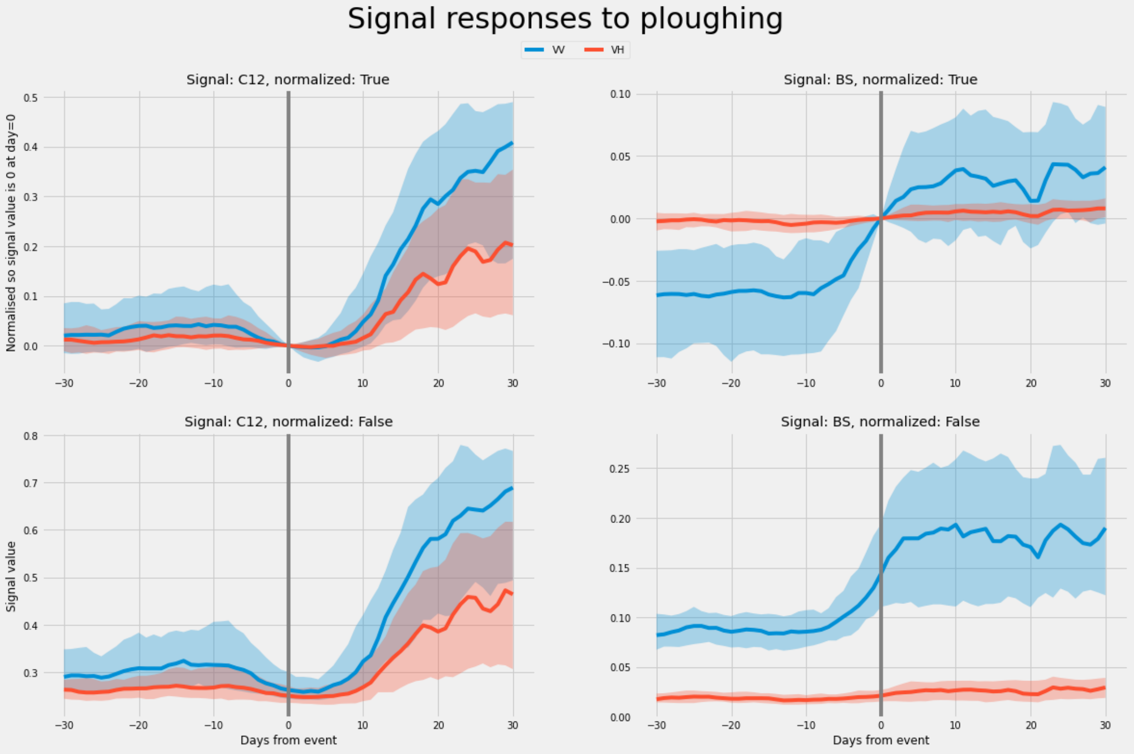

Figure 7: Median, lower and upper quartiles of extracted S1-signal medians response to ploughing event. Coherence (C12) is plotted on the left side and backscatter (BS) is plotted on the right side. Upper graphs have been normalized with respect to event time, whereas lower graphs have not been normalized. The ploughing event has occurred on day 0. Note that because of the data collection method the true event may have actually been up to seven days earlier compared to day 0 in the plot.

approach more robust to date selection we can divide the signal difference by the time difference, i.e., number of days. This leads to a formula that looks a bit like the derivative of the signal function. The larger time difference we take the smoother the derivative function looks.

By examining Figures 8 and 7 we can observe that the potential variations in the signal in future with respect to event could hold valuable information. If utilize this information we cannot be employed instantenously and instead it needs a buffer to see how the signal changes in the future. Preprocessing steps also need information from the future observations so computing the signal change in future just adds a larger buffer before we can predict the status of the parcel.

More formally: for a given parcel if we have signal value x_{t_0} at the time step t_0 then the signal difference n days backwards $d_{t_0-t_n}$ is defined as

$$d_{t_0-t_n} = \frac{x_{t_0} - x_{t_0-t_n}}{n}.$$

We can create a matrix for these values by applying this formula to all observations. This creates a matrix with a shape of $\mathbb{R}^{n\times(2\times\#\text{number of signals})}$ where we have column

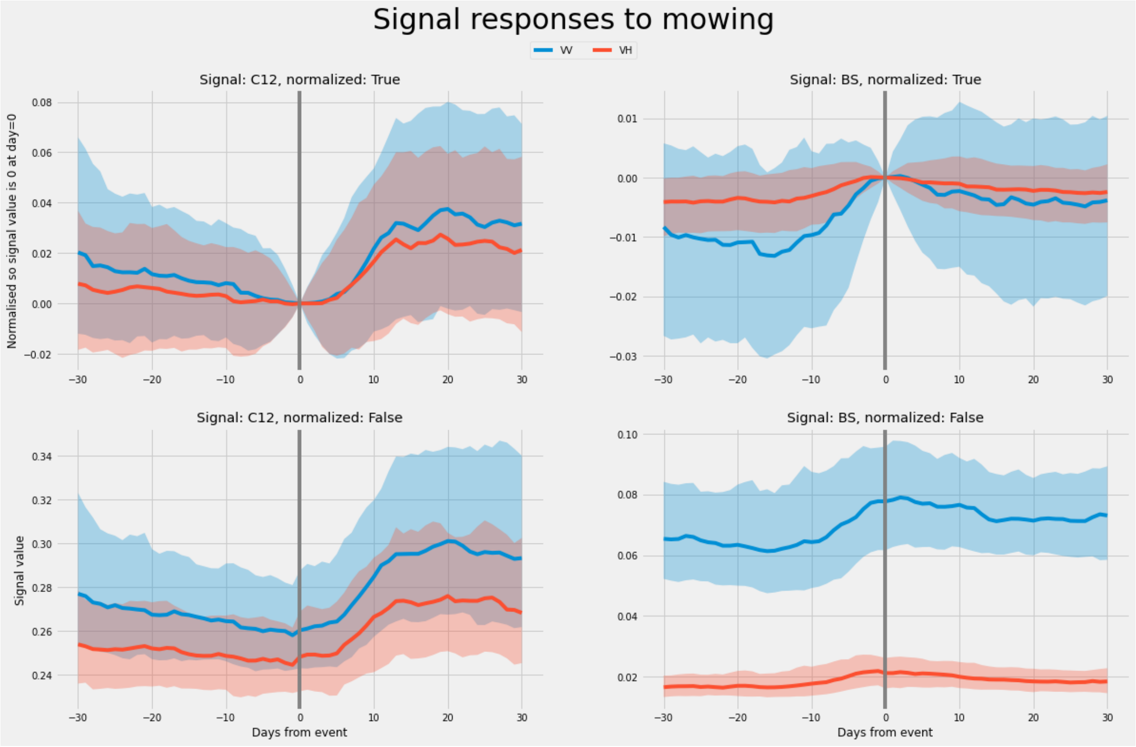

Figure 8: Same signals plotted as in the Figure 7 for the mowing event.

indicating the derivative calculated backwards and a column for the forward facing derivate for all of the signals. This operation requires more signal observations that are in the observation matrix X. X consists only from the dates that we have ground truth observations but we have signal data readily available from the longer period that is used in this study so there is no problem calculating the differences.

The time differences that we add as features in our model are part of our hyperparameters that we consider in our model and optimal values for these need to determined in our model validation. Importantly these pseudo-derivatives are calculated after the smoothing step.

The final feature which we add to our model is the information about the crop in the given parcel, i.e., grass, cereal or something else. Majority of the parcels in the dataset contain grass or cereals and the agricultural events look very different depending on the crop. The frequencies of the events are also depend on the crop type: the cereals rarely have multiple cutting events during the seasons and the grass does not ripen in the sense that cereals do. To capture this information we create a vector $C = \mathbb{R}^{n\times 2} \in \{0, 1\}$ where each row encodes the crop code information. The crop code c_i of the ith observation is derived from the crop code matrix with the formula:

$$c_i = \begin{cases} \text{Cereal} & \text{if } C_{i,1} = 1 \text{ and } C_{i,2} = 0 \\ \text{Grassland} & \text{if } C_{i,1} = 0 \text{ and } C_{i,2} = 1 \\ \text{Other crop type} & \text{if } C_{i,1} = 0 \text{ and } C_{i,2} = 0 \end{cases}, \quad i \in \{1, \ldots, n\}.$$

From now on we treat the crop type and the derivative matrices as part of feature matrix and refer them as part of the feature matrix $X \in \mathbb{R}^{n \times p}$ where n is the number of observations and p is the number of features.

At this point our dataset has the following information:

1. Parcel id
2. Date
3. Crop information
4. Processed S1 signals and S2 indices
5. The pseudo-derivative calculated forwards and backwards for all of the aforementioned signals

and the resulting number of features is

$$\begin{aligned} p &= 3 \text{ (crop code)} \\ &+ 12 \text{ (current signal values)} \\ &+ 12 \text{ (pseudo derivatives backwards)} \\ &+ 12 \text{ (pseudo derivatives backwards)} \\ &= 39. \end{aligned} \tag{6}$$

5.5 Creating the train and test splits

In order to train the model and accurately measure out of sample prediction error we need to split our dataset so that we use a part of the dataset for model creation and the remaining data to test how the model works. If we would use the same dataset in both steps the classification results would be too optimistic as the model has already seen some of the datapoints and knows the label associated with those observations [16].

Our dataset contains time series data so we can't use the naive approach of just splitting the dataset randomly because of the data leakage. For example if put in the training set the observations and the related labels from Monday and Wednesday for some parcel, the model could probably predict the correct label for Tuesday. Consequently this leads to optimistic model validation results.

The way we approach this problem in this thesis is that we split the data in to two sets by parcel ids. We could just randomly sample parcels to two sets but the sets we generate might not be representative as some of the events are less common than others. To try to make the sets representative we use the algorithm described in 1.

The algorithm works by first creating empty sets for training and test parcels and then we populate the two sets by adding parcels in a loop. We start with the parcels which contain rare events such as light tillage and move to more common events such as growing or mowing. If for example we would first divide the parcels that contain light tillage events we get a list of parcels which contain the light tillage events and shuffle those randomly. Then we start putting all the parcels in the list to either training or test set. If a parcel in the list does not belong to a training set we insert it into the test set and otherwise we skip over the parcel. We keep putting events which contain a light tillage event to test set until we have inserted half of the parcels to the test set and rest of the parcels go to the training set. Then we

Algorithm 1: Algorithm for creating training and test splits.

```
training parcels $\leftarrow \emptyset$;
test parcels $\leftarrow \emptyset$;
**for** $E \leftarrow$ Unique events ordered from least to most common **do**
    $p \leftarrow$ {parcel | parcel contains event $E$};
    $p \leftarrow$ shuffle($p$);
    total parcels containing event $\leftarrow$ length($p$);
    samples for event $\leftarrow 0$;
    **for** candidate $\leftarrow p$ **do**
        **if** candidate $\notin$ training  parcels **then**
            test parcels $\leftarrow$ test parcels $\cup$ {candidate};
             samples for event $\leftarrow$ samples for event $+ 1$;
        **end if**
        **if** samples for event $>$ total parcels containing event $/2$ **then**
            training parcels $\leftarrow$ training parcels $\cup$ ($p \setminus$ test parcels);
            Stop inner loop iteration;
        **end if**
    **end for**
**end for**
```

take a look at the parcels which contain the second most common event and repeat this procedure until we have exhausted all of the events.

Both of the resulting sets are similar in size but one could tweak the algorithm to create sets with different balances. Usually the training to test set size is about 80/20 [16] but we have relatively rare events and would like to have smaller test error bounds for those. For example only 152 parcels (about 6%) have a light tillage event.

In this analysis we have only one more rare event (light tillage) and the split in this case could have been done with random sampling and checking if both of the sets are reprenstative compared to original set. In the first iterations we had more event types and it could have been tedious to try to randomly split the dataset in to representative sets. After running the split for a fixed random seed we have 1143 parcels (109848 datapoints) in our training set and 902 parcels (90782 datapoints) in the test set. We use the training set to train the model and tweak the hyperparameters. The test set is used to measure the model error.

5.6 Predictive model

We fit the gradient boosted tree model using the training set with a one-versus-all method. In this thesis we use the gradient boosted tree model framework called XGBoost [15]. For each label (e.g. ploughing, ripening, mowing, etc.) we fit a binary classifier which produces the predicted probability for the specified label against all other labels. This leads to k different binary classifiers, where k is the number of unique events in the dataset.

Given that $X \in \mathbb{R}^{n \times p}$ is our data matrix, $y \in \mathbb{R}^{n \times 1}$ is our target variable and k

is the number of unique classes in the target vector y the classifier prediction is as follows:

$$\hat{p}_{i,j} = \hat{f}_j(X_i),$$

where $\hat{f}_j$ denotes the binary classifier for class j, X_i the features for the $i$th observation and the $\hat{p}_{i,j}$ is the predicted probability that the $i$th observation belongs to a class j. The function $\hat{f}_j$ is a mapping from the feature space into the probability space:

$$\hat{f}_j : \mathbb{R}^p \to [0,1], \quad j \in 1, \dots, k$$

The output of our classification is a $n \times k$ matrix where element $\hat{p}_{i,j}$ is the probability that the $i$th observation belongs to group k. These values have been scaled so that the rows sum to one:

$$\sum_{j=1}^{k} \hat{p}_{i,j} = 1, \quad \forall i \in 1, \dots, n.$$

The scaling of the predictions corresponds to a softmax function. One approach would have been not to scale the predictions which corresponds to a sigmoid function which is the unscaled probability. The primary difference between these two is the interpretation of the results. In the former case we are assuming there can only be one event at the time and the probabilities correspond to confidence that a specific event is happening out of all the events. In the latter case the interpretation could be "given these signals the probability corresponds this event happening". We chose to scale the predictions as the interpretation is more natural in this case as the parcel can only have one primary event at the time, e.g., the parcel can't be cut and growing at the same time.

In order to generate a prediction when the event started and when it ended we need to process these probabilities further. We approached this problem by taking the event with the highest probability and assigning it as the prediction for given day. After that we calculate most frequent event in a centered moving window and output that as prediction. The window with a mode function helps to reduce noisy predictions, i.e., events that only last a day or two.

If we have a function

$$f(\hat{p}_i) = \arg\max_j \hat{p}_{i,j},$$

which outputs the most likely prediction for the day $i$ then the smoothed prediction is defined as $\hat{y}_i = \text{mode}(f(\hat{y}_{(i-(t/2),j)}), ..., f(\hat{y}_{(i+(t/2),j)}))$, where t is length of some sliding window and mode is a function that outputs the most frequent element in a given set. Now we can define the start and end dates for the events. The start date of an event is the index $i$ where the event changes:

$$\text{Day } i \text{ is} = \begin{cases} \text{start date,} & \text{if } \hat{y}_{i-1} \neq \hat{y}_i \\ \text{end date,} & \text{if } \hat{y}_i \neq \hat{y}_{i+1}. \end{cases}$$

With the start and end dates defined we can easily count how many events happened during the observation period and compare it to ground truth and calculate the error in predicted frequencies as in Lobert et. al. [13].

5.7 Hyperparameter optimization

The preprocessing, model building and model validation takes a couple of minutes to run. For example if we tried 20 different values for each one of the days used in the derivative feature generation, 15 values for the length parameter and five different order of polynomials used in the Savitzky-Golay filter this would generate $20 \times 20 \times 15 \times 5 = 30000$ different parameter combinations. If we add the hyperparameter combinations for the k different XGBoost models the hyperparameter space is too large. Instead, we try to optimize the complete hyperparameters by first finding the optimal values for the preprocessing steps, after which we optimize the hyperparameters for the individual predictive functions $\hat{f}_j$, e.g., the XGBoost hyperparameters.

The preprocessing hyperparameters tried in the hyperparameter search are documented in Table 6. The search consisted of 300 different hyperparameter combinations. For each one of the parameter combinations the hyperparameters for the XGBoost model were the same. We logged mean error and mean absolute error 3 for all the parameter combinations and these are visualized in the Figures 9 and 10.

Parameter	Values tried
Savitzky-Golay filter length	{3, 5, 7, 11, 15}
Savitzky-Golay filter polynomial order	{1, 2, 3}
Days used when calculating derivative forwards	{2, 5, 7, 10, 15}
Days used when calculating derivative backwards	{2, 5, 10, 15}

Table 6: Different preprocessing hyperparameters used.

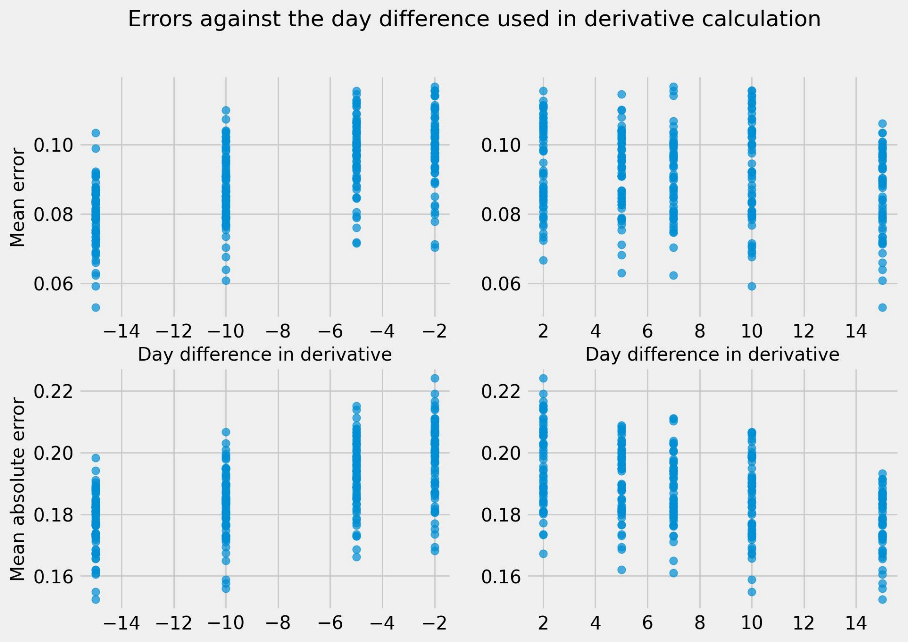

Figure 9: Errors against the day difference used when calculating the derivative.

We can see from Figure 9 that the errors generally rise the closer the day parameter in derivative calculation is relative to the event. The longer day difference we can afford to calculate the smaller the errors tend to be. The reason for this might be that the amount of smoothing applied is smaller and subsequently we get more noise and shorter events.

From Figure 10 we see that errors generally tend to be smaller when polynomial used in the filter is smaller. Higher order polynomials follow the data more accurately but some of the signals might be very noisy and in turn we get more entropy in the predictions. The results could indicate that the best parameter to use would be one. This is interesting as the Savitzky-Golay filter with polynomial order of one is just the linear unweighted mowing average filter.

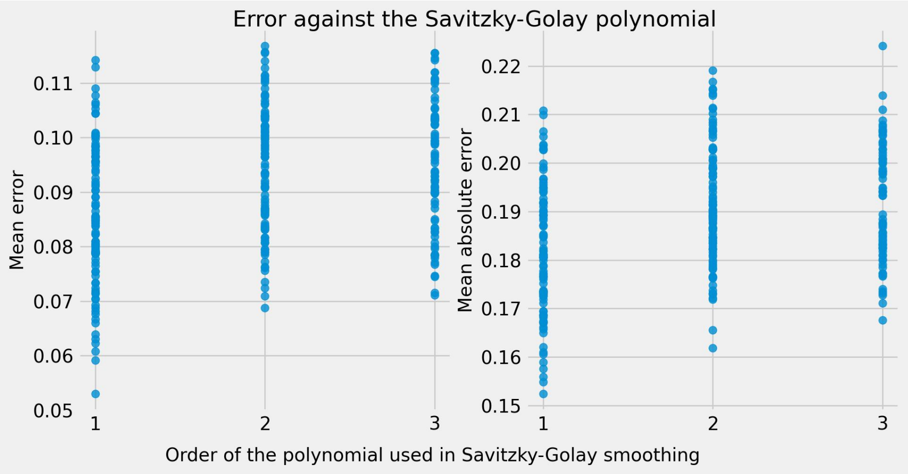

Figure 10: Error against the order of the polynomial used in Savitzky-Golay filter.

When comparing different Savitzky-Golay filter lengths all errors were very similar. When performing bootstrapping tests between the categories generally the longer filter lengths had the lower errors. Notably the model tends to overestimate all the event types as indicated by the positive ME. Generally we have more events that happen once or twice, e.g. ploughing, ripening or mowing the parcels once or twice, and less events that happen regularly, e.g., mowing the grassland parcel four times, so we might overestimate the less frequent events and underestimate the events with the higher frequency.

We are trying to run the model live with some lag so in our case we do not have unlimited time to wait for the most optimal results, so we need to pick parameter values that allow us produce results as fast as possible while maintaining accuracy. We decided to use 5 for the Savitzky-Golay filter length, 1 for the polynomial, 12 for calculating the derivative backwards and 7 for calculating the derivative forwards. The results and conclusions are derived for these parameters but do note that if you are running the model as a one time batch run the results might be better if you use other parameters.

5.8 Fitting the model

The XGBoost implementation of the gradient boosting has an enormous number of different hyperparameters and we need to train six different models. The hyperparamer space is massive and we leave it out of scope for this theses. Instead we opt mostly for the default parameters [18]. The hyperparameters which we modify is the scale_pos_weight which changes the positive label class weighting. We modify this parameter because the different classes are very unbalanced and we would like to weigh the misclassification errors with respect to their frequencies. The resulting classification report is documented in Table 8 and the confusion matrix in Table 7.

We can see from the confusion matrix that the growing and mowing gets easily confused. This is expected as the the moment when a grass parcel starts growing after it is mowed is very subjective. The parcel can look like it is growing very soon after being cut during the summer months but the satellite might not notice that for some time if the satellite does not pass over the parcel. Another event that gets mixed a lot is the light tillage class. The light tillage is quite a broad term and judging by the collected data the data gatherers are not even sure if the event is ploughing or light tillage at times. The mixup between the light tillage and the growing can be explained with the same reason that mowing and growing get mixed up. Finally the harvest and ripening get mixed up. These events are most commonly next to each other so the delays between the actual event, the time event is logged and the time event shows up in the satellite signals add up and might cause misclassifications.

		Predicted					
		Growing	Light tillage	Ploughing	Mowing	Harvest	Ripening
Actual	Growing	48640	277	202	5211	198	28
	Light tillage	472	972	511	68	202	14
	Ploughing	117	382	2231	12	189	9
	Mowing	3197	32	57	5920	0	0
	Harvest	3	115	340	0	13449	1341
	Ripening	31	7	6	0	1096	5453

Table 7: Confusion matrix for the test set.

Some of the errors show up on the regular validation metrics as seen from Table 8. Importantly we have high recall scores meaning if there is an event the algorithm finds it in most of the categories. The low recall of the mowing events is not that big of issue as they are mostly mixed with the growing event where the majority of the datapoints belong. Even if this table does not look that great the aggregated values look much better. Usually we do not care so much if we missclassify some here and there or the predicted start/end times are off by some days. We are more interested if the model correctly classifies longer periods of time to the right categories.

In Figure 11 we present the predicted probabilities $\hat{p}$ and ground truth labels for one parcel. We can see how the model correctly classsifies the alternating mowing and growing events. At the beginning and at the end of the season we have more uncertainty in the predictions as we have no ground truth data from these periods and we could not train the model at these time periods. This issue will be addressed during the data collection for summer of 2023.

	Precision	Recall	F1-score	Support
Growing	0.917	0.914	0.915	54556
Light tillage	0.625	0.427	0.507	2239
Ploughing	0.727	0.717	0.722	2940
Mowing	0.561	0.597	0.578	9206
Harvest	0.872	0.915	0.893	15248
Ripening	0.841	0.788	0.814	6593
Accuracy	0.854	0.854	0.854	0.855
Macro avg	0.757	0.726	0.738	90782
Weighted avg	0.854	0.854	0.854	90782

Table 8: Classification table for classifying the individual days computed from the test set.

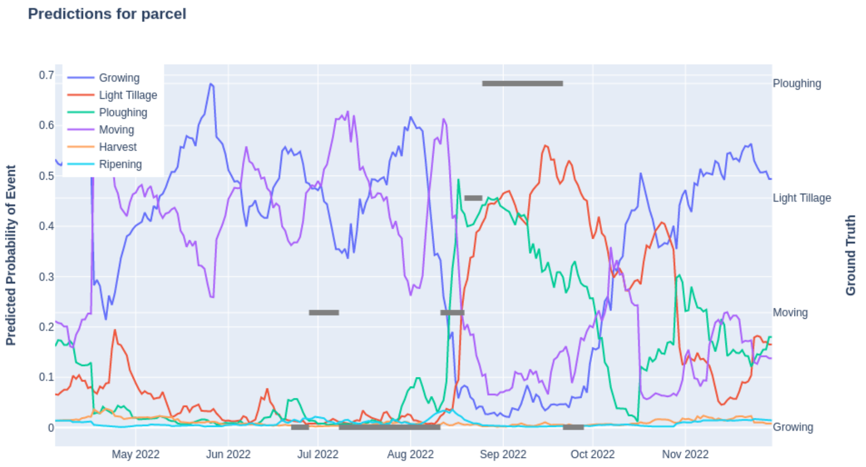

Figure 11: The predicted events for one parcel. Left y-axis is the predicted probability $\hat{p}$ and in the right y-axis there is the observed event from ground truth.

5.9 Comparing aggregated predictions

Now we have true frequencies of the events from the ground truth data and the predicted frequencies by using the methods presented in Section 5.6. By using the predicted start and end times we can classify individual days to periods of time when the parcel had one event. We can compare the total event counts of these two by using chi-squared test [19] and see if the model learned the true distribution of the frequencies. Let our null hypothesis H0 be: the frequencies of the events in true and predicted categories are same. The alternative hypothesis HA is that the frequencies in the groups are different. Set the confidence level to 99% which corresponds to p-value of 0.01. Using the frequencies from Table 9 we get the value of the t-test statistic $\chi^2 = 23.971$ and the corresponding p-value is 0.00023. This means we can reject the null-hypothesis and conclude that model did not learn the true frequencies.

When we examine Table 9 it becomes apparent that the majority of errors stem from an overprediction of Growing events. With the exception of the aforementioned issue, the model is accurate at predicting the other event types. Usually the Growing is not the primary event of interest and we are more interested on the active events so it is more important for the model to spot the other events.

Event	True count	Predicted count
Growing	855	1082
Light tillage	73	70
Ploughing	163	166
Mowing	733	663
Harvest	351	377
Ripening	345	371
Total	2520	2757

Table 9: Total event counts compared to predicted event counts.

The mean errors, mean absolute error and normalized mean absolute errors are documented in Table 10. The metrics are also calculated individually for the Cereals and Grass parcels as some of the events only belong to either type.

		Light Tillage	Ploughing	Mowing	Harvest	Ripenining
Cereals	ME	-0.037	0.014	0.00	0.032	0.063
	MAE	0.077	0.112	0.00	0.193	0.092
	nMAE	-0.421	-0.117	-	0.005	0.03
Grass parcels	ME	0.019	-0.009	-0.145	0.000	0.00
	MAE	0.064	0.036	0.341	0.000	0.00
	nMAE	-0.304	-0.200	-0.054	-	-
All parcels	ME	-0.003	0.003	-0.085	0.024	0.022
	MAE	0.074	0.069	0.200	0.086	0.039
	nMAE	-0.396	-0.147	-0.054	0.016	0.026

Table 10: Total event counts compared to predicted event counts. Dashes indicate the value could not be calculated as the delimiter in the formula is zero (in the ground truth dataset there are no observations for that specific event).

The highest errors correspond to underestimating the amount of mowing events on grass parcels as indicated by the negative mean error. One reason for this might be the lack of optical (S2) signals. The model works even with large periods of missing signals because of the interpolation but if we don't receive any signals for a long period time we might miss the changes that would indicate the mowing events. The mowing events are primarily detected from changes to optical signals as seen from Table 13 and if we cannot record these changes the accuracy suffers. Solutions to this problem is furher discussed in Section 6.1. The feature importances presented are computed by averaging gain across all splits in all of the trees where the feature is used in. If the features are very correlated the the feature importance might not be credible as the. In this case the signals have high correlations so the credibility of the feature importances should be taken with a grain of salt.

One major thing missing from Table 10 is the information whetever the predictions are correctly predicted at the correct times. There is some subjectivity what counts as a "correctly predicted". Should the predicted events start and end at the same date as the correct ones? Should there be some overlap to be correctly predicted? How long overlap? We decided to drop this analysis from this thesis as these decisions influence the metrics wildly.

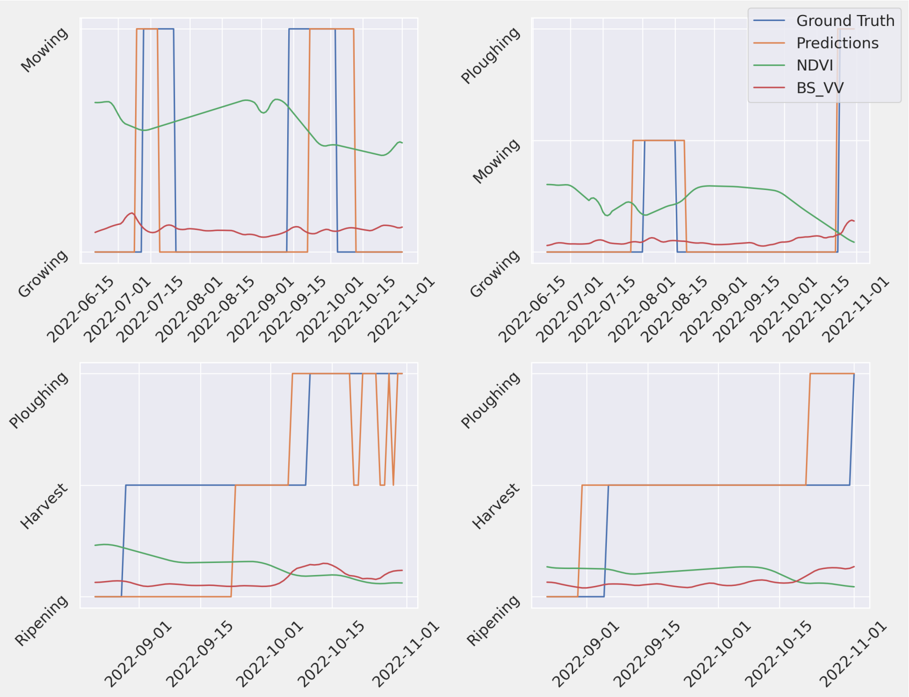

Figure 12: Examples of model predictions. Dropping NDVI values indicate there is less vegatation in the parcel which could mean there is a mowing or ripening event. The rising backscatter could indicate the there is more bare soil which is an indication of ploughing or maybe light tillage.

Visually inspecting the predictions it seems like the model correctly predicts the events next to the actual events. If we take a look at Figure 12 from the top left corner that the model correctly finds the two mowing events although the start time of the second event is off by some time. In the bottom left corner we see that our prediction smoothing filter, described at the end of Section 5.6 does not correctly work and we end up with quite a few extra predictions for the harvest and the ripening is placed few weeks after the actual ripening.

6 Discussion

6.1 Improvements

We noticed that one reason for missing events is the irregular optical data acquisition. The reason for the irregular data is cloudiness. Sentinel-2 constellations revisits certain location every five days [3]. This is very high frequency compared to other optical satellites such as Landsat-7 that has a revisit period of 16 days [20]. But if we happen to have couple of cloudy acquisition days back to back we might easily miss some events. For example during the summer months the grass grows very quickly.

This uncertainty is not seen in the predictions. One solution would be to not output predictions if there is too much time between signal acquisitions. If we happen to have a very cloudy summer this makes the monitoring very hard if we can't make predictions for a long time.

Another solution is to replace some of the optical signals with the radar counterparts. For example there have been studies to approximate NDVI from the radar signals such as study done by Pelta et. al. [21] where they try to predict the NDVI using the radar signals or to replace the NDVI altogether with the radar vegetation index introduced in the study done by Sahadevan et. al. [22]. It is important to note that that the analysis was limited to 12-day coherence, and getting more accurate data might make a big difference.

For now we are using all of the variables in our analysis. We have quite a few indices and radar signals. When we calculate the current value, derivative forwards and backwards we end up with a lot of of variables. Some of these variables could be pruned using the feature importances that you can extract from the gradient boosting model. Our feature importances are presented in Appendix and we can clearly see that for some models the feature importances are very low. To prune some variables we would have to find variables that low importance score across all of the models. The XGBoost is quite robust to low information variables but the best practice would be to be prune the unnecessary variables.

Currently the model assumes that there is only primary status at the time. The model is easy to train and the results are easy to interpret but if one would like to include grazing event in the model, which usually occurs when the grassland parcel is either growing or mowed, we run in to more issues. When there is a grazing, which usually looks like a growing event, the model does not perform greatly in our test cases. One issue in this study is the low number of examples from the grazing events which might play a role. We decided to drop the grazing from the list of possible events but we might have to return to modelling that later on.

Another thing one might notice from the list of events is that none of the events fit in to early summer for the cereals. The seeds have been recently planted but the plant is not yet ripe. The data collection for the cereals in the summer of 2022 began when the cereals ripened so we have no records on any other types of events before that. The data acquisitions starts sooner during the next summer with the goal of being able to monitor the parcels all the way from early spring until the late fall.

This bring us in to the final improvement: time. The model does not know if it is early spring, cold winter or a late autumn. Adding the time as a feature into the model did not produce high improvements into the accuracy. This could be explained by the fact that most of the data is from one or two months. In fact the model does not know what the previous predicted event was. It is not physically possible for cereals to first be harvested and then ripen. Because the model does not take the order of events in to the account it is possible for model to output these kinds of predictions. One possibility is to change the gradient boosted tree model into a Hidden Markov Model [23] which considers the probability of transitioning from one state to another.

6.2 Conclusions

In this thesis we first introduced different satellites, various signals derived from the satellite imaginery and associated terminology. After this we did a brief literature review and applied some suggestions into our own model. The metrics proposed in Lobert et. al. [13] are especially useful when comparing the counts of various predictions.

After this we took a deep dive into the inner workings of gradient boosting methods and signal processing methods. We then applied the various methods and created a preprocessing method which created a good baseline from which we could start building different classifiers. After applying a classifier to the processed signals we created a method to detect various agricultural activities from the satellite signal time series.

Overall the method proposed in this thesis works very well even if the labels derived for each day from the weekly visits might be inaccurate. The validation results indicate that errors between this method and the one proposed by Lobert et. al. are similar. Finally we took a look at the various challenges, inaccuracies and propose improvements to furher improve the model.

The method developed in this thesis will be tested in Finland during the summer of 2023 and if the results prove to be succesful, additional events may be added into the model in the future. However, the code to produce results is not unfortunately open source. Although there is a a possibility that the model and analyses might become available for the public use at the later time, but for the time being the code can not be freely accessed.

Machine learning and deep learning techniques have the potential to greatly enhance the capabilities of remote sensing technology in the future. One of the main advantages of these approaches is that they can be used to automatically extract patterns and recognize changes from large and complex datasets such as datasets generated by satellites and drones.

Satellites generate vast quantities of data and automated and efficient processing of this data can help reduce the time and resources to extract meaningful insights from these remote sensing datasets. Using remote sensing technology, we can obtain up-to-date information on the land cover of the earth which can be used to derive valuable insights about our planet. For example remote sensing can be used to monitor and track changes in natural systems such as forests, rainforests and glaciers. Or remote sensing can be used in assessing the human impact on the planet; for example we can track urbanization, air pollution, agriculture or even the frontlines in the war in Ukraine.

Automating these remote sensing workflows using machine learning techiques we can speed up the response to many natural disasters such as floods, wildfires and earth quakes. Using the satellites or drones in disaster areas can save human lives by providing accurate information. [24]

While the remote sensing and machine learning have the potential to provide many helpful benefits to society they can also be used in harmful manner. For example using high resolution satellite imagery it is possible to track boats [25] or planes [26] in a airfield and by using drones it is even possible to track individual humans [27]. These techniques could be used to facilitate mass surveillance. In the war in Ukraine we have already seen these techniques being used. Consumer-grade drones are dropping grenades on soldiers [28] and high resolution satellite imagery is tracking movements of military equipment and batallions [29]. Taking the human out of this equation might lead to a new levels of automation and increase the scale and scope of violence.

Ultimately machine learning and remote sensing have potential to provide numerous benefits to human society by revolutionizing our understanding of natural and urban environments. These sources of information can help us make informed decisions and polices to enchance sustainability of life on earth but it is crucial to consider potential negative impacts and establish legal frameworks to ensure these technologies are used in responsible and ethical manner.

References

[1] *Sentinels for Common Agricultural Policy - Sen4CAP*, http://esa-sen4cap.org/content/project-background, Accessed 22.11.2022.

[2] *Sentinel-1*, https://sentinel.esa.int/web/sentinel/missions/sentinel-1, Accessed 22.11.2022 .

[3] *Sentinel-2 MSI Technical Guide*, https://sentinels.copernicus.eu/web/sentinel/technical-guides/sentinel-2-msi, Accessed, 26.1.2022.

[4] *Learn Synthetic Aperture Radar (SAR) by Example*, https://gisgeography.com/synthetic-aperture-radar-examples/, Accessed 21.12.2022.

[5] *What is Synthetic Aperture Radar?*, https://www.earthdata.nasa.gov/learn/backgrounders/what-is-sar, Accessed 3.1.2022.

[6] Closson, D.; Milisavljevic N. *InSAR Coherence and Intensity Changes Detection*, InTechOpen, 2017, http://dx.doi.org/10.5772/65779.

[7] *Mission ends for Copernicus Sentinel-1B satellite*, https://www.esa.int/Applications/Observing_the_Earth/Copernicus/Sentinel-1/Mission_ends_for_Copernicus_Sentinel-1B_satellite, Accessed, 24.4.2023.

[8] *Sentinel-1 Observation Scenario*, https://sentinel.esa.int/web/sentinel/missions/sentinel-1/observation-scenario, Accessed 30.12.2022.

[9] Sica, F.; Pulella, A.; Nannini, M.; Pinheiro, M.; Rizzoli, P. *Repeat-pass SAR interferometry for land cover classification: A methodology using Sentinel-1 Short-Time-Series*, Remote Sensing of Environment, 2019, http://dx.doi.org/10.1016/j.rse.2019.111277.

[10] *Index Database: a Database for Remote Sensing Indices*, https://www.indexdatabase.de/db/s-single.php?id=96, Accessed 25.4.2023.

[11] Chang, K. *Introduction to Geographic information systems*, McGraw Hill, 2018, ISBN10: 1259929647.

[12] Voormansik, K.; Zalite, K.; Sünter, I.; Tamm, T.; Koppel, K.; Verro, T.; Brauns, A.; Jakovels, D.; Praks, J. *Separability of Mowing and Ploughing Events on Short Temporal Baseline Sentinel-1 Coherence Time Series*, Remote Sensing, 2020, https://doi.org/10.3390/rs12223784.

[13] Lobert, F.; Holtgrave, A.K.; Schwieder, M.; Pause, M.; Vogt, J.; Gocht, A.; Erasmi, S.*Mowing event detection in permanent grasslands: Systematic evaluation of input features from Sentinel-1, Sentinel-2, and Landsat 8 time series*, Remote Sensing, 2021, https://doi.org/10.1016/j.rse.2021.112751.

[14] LeCun, Y.; Bengio, Y.; Hinton, G. *Deep Learning*, Nature, 2015, https://doi.org/10.1038/nature14539.

[15] Chen, T.; Guestrin, C.*XGBoost: A Scalable Tree Boosting System*, ACM, 2016, https://doi.org/10.1145%2F2939672.2939785.

[16] Hastie, T.; Tibshirani, R.; Friedman, J.H. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer series in statistics, 2009, ISBN:9780387848846.

[17] Savitzky, A;, Golay, M.J.E, *Smoothing and differentiation of data by simplified least squares procedures*, Analytical Chemistry, 1964, https://doi.org/10.1021/ac60214a047.

[18] *XGBoost Parameters*, https://xgboost.readthedocs.io/en/stable/parameter.html, Accessed 21.3.2022.

[19] Pearson, K. *On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random.* Philosopchical Magazine Series 1, 1900, https://doi.org/10.1080/14786440009463897.

[20] *Landsat 7*, https://landsat.gsfc.nasa.gov/satellites/landsat-7/, Accessed 16.3.2023.

[21] Pelta, R.; Beeri, O.; Tarshish, R.; Shilo, T. *Sentinel-1 to NDVI for Agricultural Fields Using Hyperlocal Dynamic Machine Learning Approach*, Water, 2022, https://doi.org/10.3390/w14111676.

[22] Sahadevan, D.; Sitiraju, S.; Sharma, J. (2013) *Radar Vegetation Index as an Alternative to NDVI for Monitoring of Soyabean and Cotton*, Remote Sensing, 2022, https://doi.org/10.3390/rs14112600.

[23] Rabiner, R.; Juang, B.*An introduction to hidden Markov models*, IEEE ASSP Magazine, 1986, https://doi.org/10.1109/MASSP.1986.1165342.

[24] Boccardo, P.; Tonolo, F. *Remote sensing role in emergency mapping for disaster response*, Springer International Publishing, 2015, https://doi.org/10.1007/978-3-319-09048-1_3.

[25] Elvidge, C.D.; Zhizhin, M.; Baugh, K.; Hsu, F.-C. *Automatic Boat Identification System for VIIRS Low Light Imaging Data*, Remote Sensing, 2015, https://doi.org/10.3390/rs70303020.

[26] Wu, Q.; Sun H.; Sun, X.; Zhang, D.; Fu, K; Wang, H. *Aircraft Recognition in High-Resolution Optical Satellite Remote Sensing Images*, IEEE Geoscience and Remote Sensing Letters, 2015, https://doi.org/10.1109/LGRS.2014.2328358.

[27] Lin, Y.; Wang, M.; Chen, W.; Gao, W.; Li, L.; Liu, Y. *Multiple Object Tracking of Drone Videos by a Temporal-Association Network with Separated-Tasks Structure*, Remote Sensing, 2022, https://doi.org/10.3390/rs14163862.

[28] Kunertova, D. *The war in Ukraine shows the game-changing effect of drones depends on the game*, Bulletin of the Atomic Scientists, 2023, `https://doi.org/10.1080/00963402.2023.2178180`.

[29] Werner, D. *Ukraine gains enhanced access to Iceye imagery and data* (News article), SpaceNews, 2022, `https://spacenews.com/iceye-ukraine-sar/`, Accessed 25.4.2023.

Appendix: Feature importances

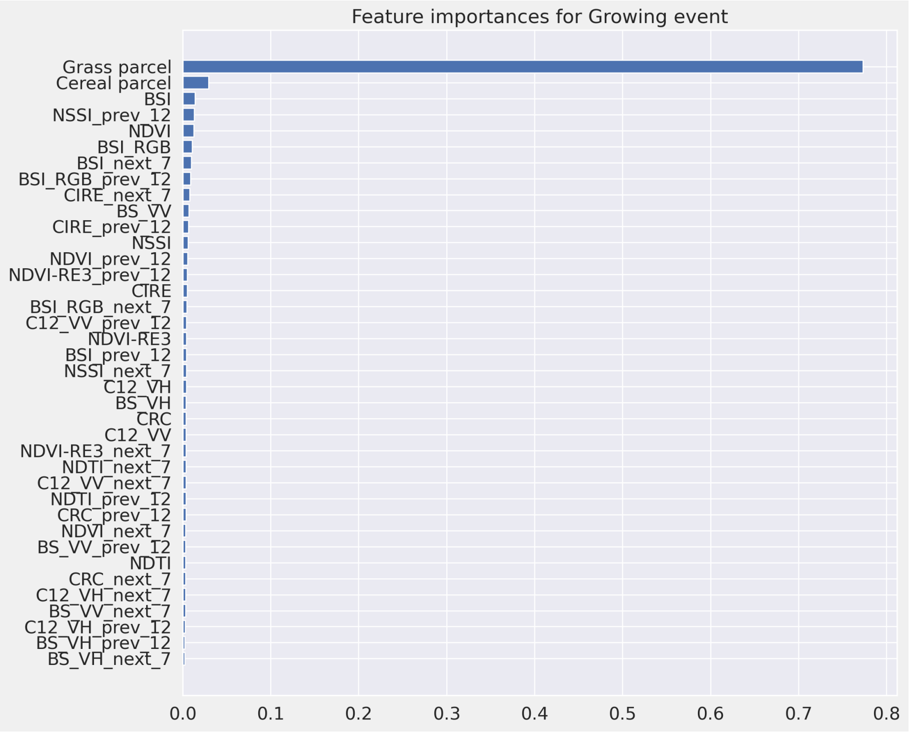

Figure 13: Feature importances for Growing event

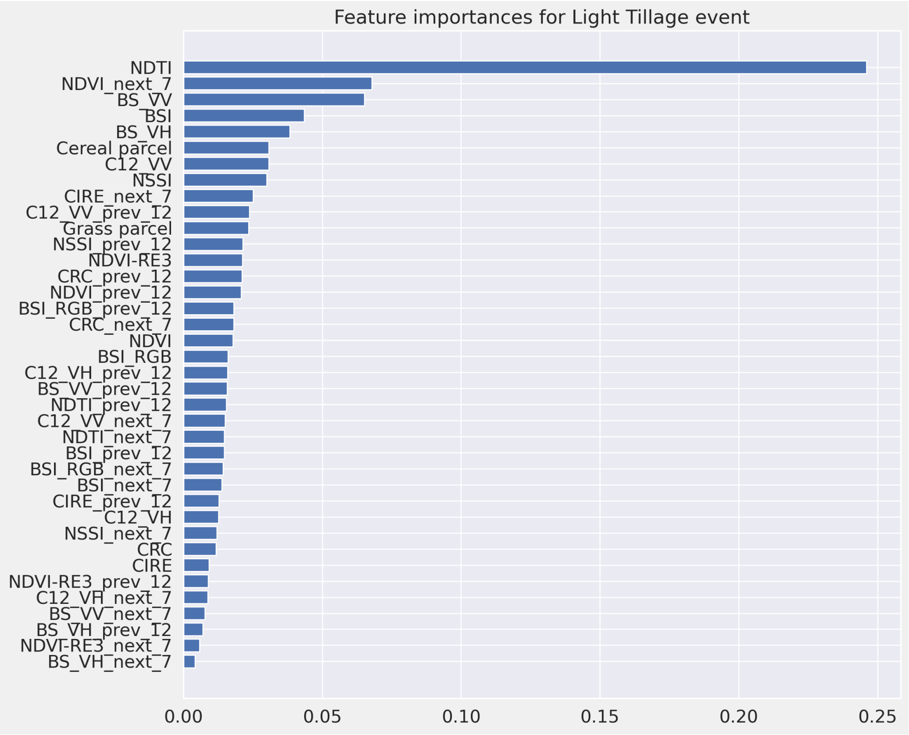

Figure 14: Feature importances for Light Tillage event

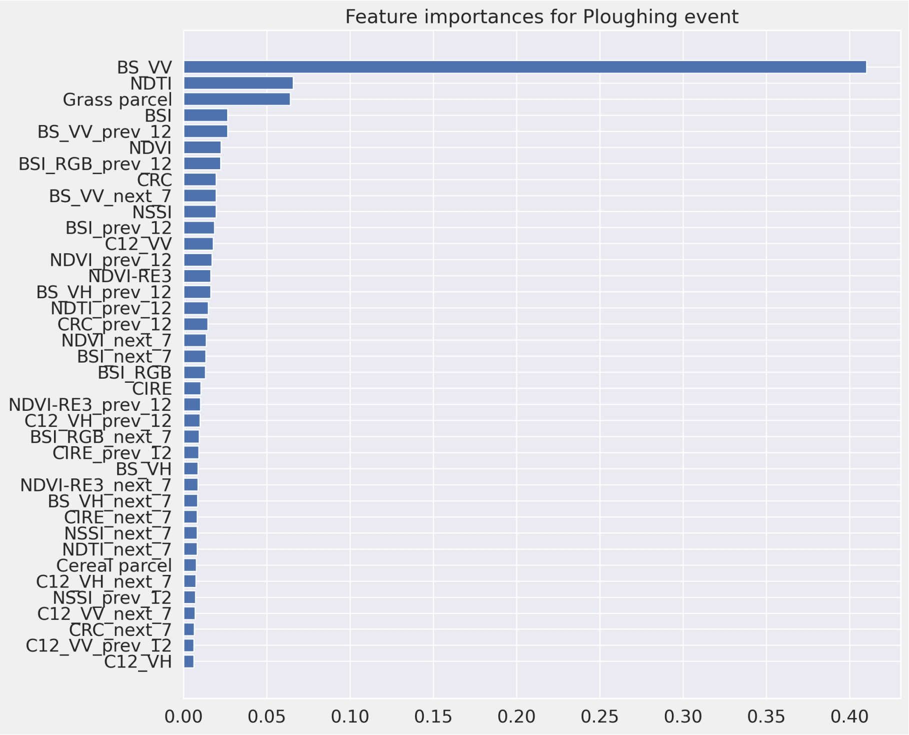

Figure 15: Feature importances for Ploughing event

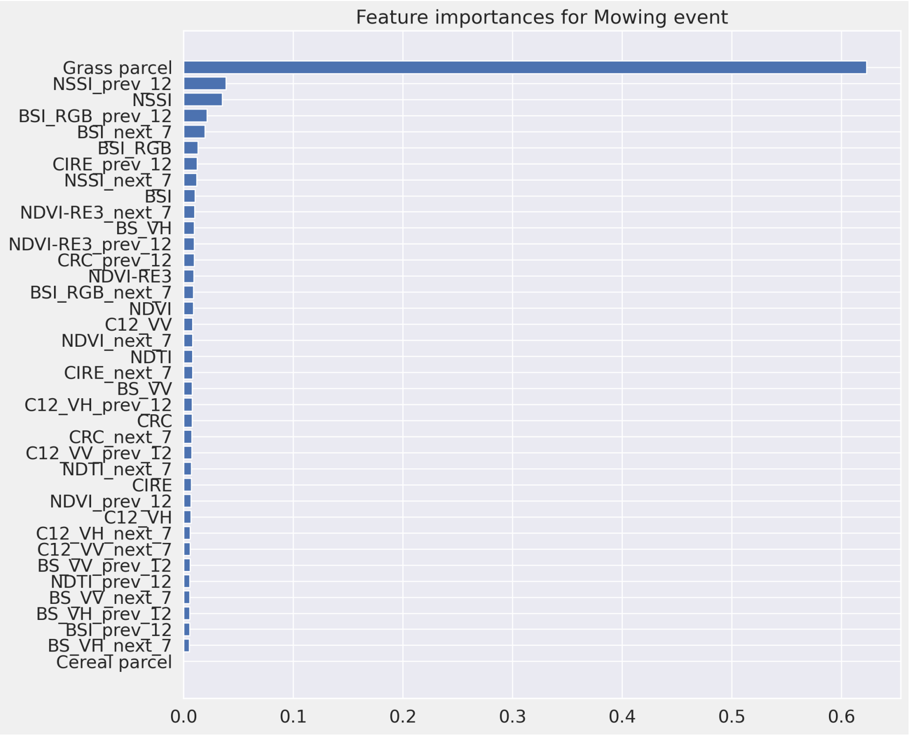

Figure 16: Feature importances for Mowing event

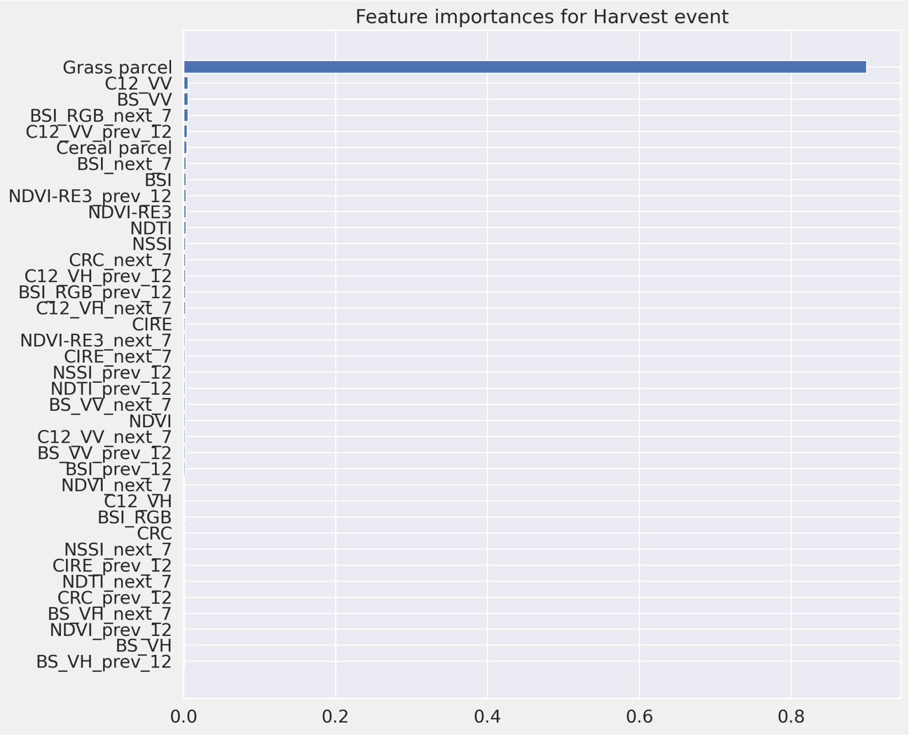

Figure 17: Feature importances for Harvest event

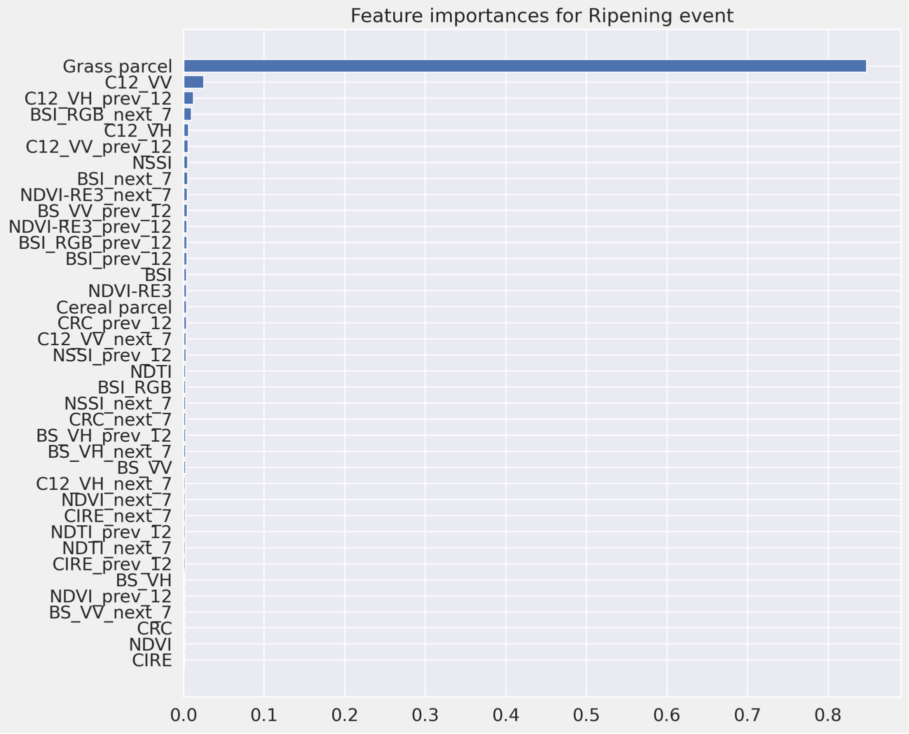

Figure 18: Feature importances for Ripening event