Master Thesis in Geographical Information Science nr 170

Land cover classification using machine-learning techniques applied to fused multi-modal satellite imagery and time series data

Anastasia Sarelli

2024
Department of
Physical Geography and Ecosystem Science
Centre for Geographical Information Systems
Lund University
Sölvegatan 12
S-223 62 Lund
Sweden

Anastasia Sarelli (2024). Land cover classification using machine-learning techniques applied to fused multi-modal satellite imagery and time series data.

Master degree thesis, 30/ credits in Master in Geographical Information Science
Department of Physical Geography and Ecosystem Science, Lund University

Abstract

Land cover classification is one of the most studied topics in the field of remote sensing, involving the use of data from satellite sensors to analyze and categorize different land surface types. There are numerous satellite products available, each offering different spatial, spectral, and temporal resolutions. Consequently, several methodologies have been developed to efficiently determine land cover using remote sensing imagery according to the spectral characteristics of each land cover category.

The objective of this thesis is to classify an area located in the Ionian region of Greece, identifying *'Artificial', 'Bare Soil', 'Cropland', 'Dense Forest', 'Grassland', 'Low-density Urban', 'Low/Sparse Vegetation*, and *'Water'* classes. To do so, the study investigates the performance of different techniques for processing and integrating remote sensing data obtained from various sensors. Multi-spectral and thermal imagery are employed, as well as topographic data from the area of interest. Landsat 8 and Landsat 9 images were specifically chosen for this project, as they include both multi-spectral and thermal information in a single acquisition. Additionally, ASTER GDEM data was used for elevation information and the generation of two elevation derivatives, the aspect and the slope of the study area. These factors, along with their temporal variability, are considered crucial as the spectral properties of certain key classes (specifically those related to vegetation and agricultural activities) are influenced by the phenological cycle.

The study addresses several research questions, including the impact of thermal information, elevation, and topography on the classification accuracy, as well as the utilization of time series data to enhance the results compared to using only the multispectral information as input. The findings indicate that combining multi-spectral data with either terrain information, thermal infrared bands, or both, significantly improves the classification results using both k-Nearest Neighbor and Random Forests classifiers. The highest performance in classification accuracy is achieved when incorporating the time series information of all the aforementioned factors as input to the Random Forests classifier. This integration yields improvements of up to 68% in specific classes, primarily those associated with vegetation.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ANN	Artificial Neural Network
AOI	Area of Interest
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer
BT	Brightness Temperature
CA	Classification Approach
CLC	CORINE Land Cover
DEM	Digital Elevation Model
DN	Digital Number
DT	Decision Tree
EPSG	European Petroleum Survey Group
ESA	European Space Agency
GDEM	Global Digital Elevation Model
JPL	Jet Propulsion Laboratory
kNN	kNearest Neighbor
LCC	Land Cover Classification
LULC	Land Use/Land Cover
ML	Machine Learning
MS	Multispectral
NASA/USGS	National Aeronautics and Space Administration / United States Geological Survey
NDVI	Normalized Difference Vegetation Index
NIR	Near Infrared
NOAA	National Oceanic and Atmospheric Administration
OA	Overall Accuracy
OLI	Operational Land Imager
FCC	False Color Composite
QGIS	Quantum Geographic Information System
RGB	Red-Green-Blue
SVM	Support Vector Machine
SWIR	Shortwave Infrared
TCC	True Color Composite
TIRS	Thermal Infrared Sensor
TOA	Top of Atmosphere
TPI	Topographic Position Index
UTM	Universal Transverse Mercator
WGS	World Geodetic System

1. Introduction

Land cover classification is a multidimensional problem. To deal with the existing landscape complexity, either natural or man-made, the photo-interpreter needs to identify and define thematic classes, by acquiring knowledge through photo-interpretation, statistical analysis, literature review, and personal experience of the phenomenon. This, however, introduces a semantic gap between the high-order semantics used by the experts in such class definitions - e.g. qualitative descriptions such as "dense forest," "urban area," or "wetland", and the low-level data-driven numeric information often in the form of digital numbers and pixel values. Bridging this gap requires the investigation and implementation of algorithms and models that can effectively utilize the numerical data to extract human-meaningful land cover classifications. To this end, existing studies investigated the (semi)automation of land use/cover classification by examining the potential of the application of machine learning algorithms to extract thematic categories from multi-modal data.

Liya and Schulz (2015) propose that the combination of multispectral indices along with thermal band information via time series analysis of at least 5 or 6 thermal images significantly improves the land cover classification results, compared to using only standard VIS/NIR bands. Gounaridis, Apostolou, and Koukoulas (2016) found that areas covered with vegetation had the highest inaccuracies due to variations of vegetation characteristics as a function of the phenological cycle. These classes are referred to as 'Heterogeneous agricultural areas', 'Permanent crops', 'Scrub and/or herbaceous vegetation associations', and 'Forests' of the CORINE Land Cover 2000. Liu et al. (2018) suggest that the highest accuracy (82.78%) can be achieved with fused terrain and multi-temporal multispectral data for the identification of forest types.

Both Simonetti, Simonetti, and Preatoni (2014) and Schäfer et al. (2019) studied the temporal aspect of the land cover classification procedure. Even if these algorithms that exploit the periodic changes over pixel time series of medium-resolution satellite imagery are a very recent innovation in the scientific community (Hostert et al, 2015), the findings are promising. The first study achieved an overall accuracy of 89.9% through a time series analysis procedure over a mountainous area with a variety of vegetation types. The latter which used, among others, the Random Forests (RF) machine learning methods also achieved high overall accuracy in the land cover classification procedure (88.7%) for a total of 9 different classes. The selected classes were namely the: 'Urban Areas', 'Other built-up surfaces', 'Forests', 'Sparse Vegetation', 'Rocks and Bare Soil', 'Grassland', 'Sugarcane crops', 'Other crops', and 'Water' over a study area in the Reunion Island, France. In general, medium-resolution time series data have been employed to document forest disturbance, as demonstrated by Kennedy et al. in 2010, and to identify surface water bodies, as highlighted by Tulbure and Broich in 2013. Furthermore, it has been used to characterize changes in land cover (Zhu and Woodcock, 2014) and to identify the specifics of such land cover alterations (Olthof and Fraser, 2014).

Talukdar et al. (2020) examined the application of Random Forests, Support Vector Machine (SVM), Artificial Neural Network (ANN), as well as other ML algorithms for Land Use / Land Cover (LULC) classification using single-date Landsat 8 imagery. The study area was the river Ganga from Rajmahal to Farakka barrage in India and the studied classes were the: 'Water Body', 'Sandbar', 'Built-up area', 'Vegetation', 'Fallow land'', and 'Agricultural Land'. Random Forest achieved the highest Kappa coefficient score (0.89). Hosseiny et al. (2022) used more data inputs, including terrain information, vegetation indices, as well as land surface phenology, and image texture information in combination with Sentinel-2 multispectral imagery to extract better accuracy. Among the studied algorithms, their results from the RF model that showed the best classification performance was the one that incorporated all the abovementioned datasets (overall accuracy = 83%, Kappa = 0.81). Svoboda et al. (2022) applied RF for land use / land cover classification from Sentinel-2 data. Having as area of study regions in the Czech Republic, the selected classes were the 'Settlements', 'Cropland', 'Grassland', 'Forest land', and 'Wetlands'. Classification achieved a high accuracy (89.1% overall accuracy, 0.84 Kappa coefficient). Thakur and Panse (2022) investigated the application of the Decision Tree (DT), kNearest Neighbor (kNN), SVM, and RF for land cover classification. Data used included the 13 bands of 27,000 Sentinel-2 images (64x64 pixels) included in the EuroSAT dataset. The classes defined for the classification process were the 'Annual Crop', 'Forest', 'Herbaceous Vegetation', 'Highway', 'Industrial', 'Pasture', 'Permanent Crop', 'Residential', 'River', and 'Sea Lake'. Results showed that RF provided better results when compared to the other approaches (94.4% producer's accuracy). Yuh et al. (2023) examined kNN, SVM, ANN, and RF to identify Land Use/Cover changes in the Mayo Rey department of North Province, Cameroon. Data used included the multispectral bands from a Landsat 7 ETM+ imagery acquired in November 2000 and a Landsat 8 imagery acquired in November 2020. Samples were acquired for the Croplands, Dense Forest, Grassland savanna, Open savanna/ barelands, Built-up areas, Water bodies, Wetlands, Woody savanna classes. All algorithms showed satisfactory results, with RF providing the best result (Kappa statistics 94%).

To summarize all the above, previous studies that incorporated machine learning for land cover classification have examined thematic categories corresponding each time to the task at hand, to properly model and describe the region of interest. Furthermore, it has been shown that there is potential in the integration of additional dataset types such as thermal data and terrain-related indices.

Thus, this study investigates the issue of land cover classification, in the region of Ionian Islands in Greece, by bridging the semantic gap between the high order expert semantics and the low-level numerical information, through state-of-the-art supervised Machine Learning (ML) techniques and multi-modal datasets. The datasets employed in this study involved multispectral imagery, topography, and thermal information describing different aspects of the land surface. Time series analysis was also investigated to take advantage of. seasonality which plays an essential role in the spectral properties of some key classes (i.e. vegetation and agricultural-related categories).

The knowledge gap addressed in the present thesis is to test the utility of a more complex classification method that has as input a larger variety of datasets for the land cover type estimation than it is most often used. Hence, the addressed scientific problem includes these two topics:

- Aggregation to the classes' multispectral (MS) properties of information related to its thermal properties and the area's terrain elevation.
- The usage of kNearest Keighbor and Random Forests machine learning algorithms for the implementation of the land cover classification algorithm.

The research questions are focused on the selected area of interest and will be the following:

- Does the integration of the surface's thermal information with MS data improve classification results?
- Does the integration of the terrain's topography information with MS data improve classification results?
- Does the combination of all the above information improve classification results?
- Does the usage of the time series of all the above information improve classification results?

All of the above questions refer not only to the overall classification result, but also to the level of performance (User Accuracy, Producer Accuracy, Kappa) of each of the studied land cover classes.

To address the research questions, five classification approaches were tested:

1. Using as input only the MS imagery (reference)
2. Using as input both MS and thermal information
3. Using as input both MS and terrain information
4. Using as input MS, thermal and terrain information
5. Using as input the time series information of all of the above - MS, thermal and terrain.

In this way, the performance of each classification is calculated for the selected study area and compared with the rest of the classification outputs. Both per-class and overall classification accuracies will be produced, but kappa statistics (overall and per-class) will also be used for the evaluation of results to balance the potential effects of user and producer errors.

2. Literature Review

Land cover classification from multispectral satellite imagery is one of the most studied topics in the field of remote sensing. Since it benefits from various operational satellite sensors offering diverse products in terms of resolution and spectral characteristics, multiple methodologies aim to effectively classify land cover using remote sensing imagery tailored to each class's spectral features.

In this thesis, a classification approach for land cover class types over a specific area of interest is applied, along with the use of a variety of satellite datasets as input into a machine learning pipeline. Many remote sensing sensors capture information over several ranges of wavelengths within the electromagnetic spectrum, providing the scientific community with a free-of-charge, valuable means of research in fields that demand a large number of datasets, as is the case in the problem of land cover classification. Furthermore, this information is in raster form, with coverage available such that even areas with rough terrain that are difficult to reach for in-situ fieldwork can be analyzed for land cover determination. Hence, remote sensing technology can complement traditional methods while at the same time reducing the cost of fieldwork and time.

2.1. Input datasets

In the last few decades, more and more satellites have been launched, acquiring large volumes of satellite imagery with global coverage. This, coupled with free data availability, has allowed access to large volumes of current and historical data to aid research on the use of multispectral imagery as a primary input for LULC modeling (Sohl et al., 2012; Campbell et al, 2011).

Two main sources of free multispectral images are the National Aeronautics and Space Administration / United States Geological Survey (NASA/USGS) Landsat Program and the European Space Agency (ESA) Copernicus Sentinel-2 mission. Landsat imagery has been available since the early 1970s and has been commonly used for LULC classification with varying degrees of success (Amini et al., 2022; Phiri and Morgenroth, 2017; Yuan et al., 2005). Landsat 8 and Landsat 9 are the two latest missions of the Landsat program and provide multispectral imagery in the visible, near-infrared, and short-wavelength infrared spectra at a spatial resolution of 30 meters with a 16-day recurrence interval, and thermal infrared imagery, which is useful in providing more accurate surface temperatures and is collected at 100 meters (USGS, 2023).

ESA's Sentinel-2 constellation launched in 2015 and 2017 (Sentinel-2A and 2B respectively) and provides imagery at finer spatial resolution (10 and 20 m), shorter repetition intervals (5 days), and also with improved spectral resolution (three red-edge spectral bands of vegetation in addition to the visible, near-infrared and short-wavelength infrared bands) (ESA, 2023). These improvements have given the research community free access to high-quality images specifically designed for vegetation studies. This leads to an increase in the overall accuracy of LULC classification-including crop classification (Forkuor et al., 2018; Sánchez et al., 2022) at the expense of increasing data size and computational costs.

2.2. Land Cover Classification Machine Learning Algorithms

Samuel (1959) defined Machine Learning as *the field of study that provides computers the ability to* *learn without being explicitly programmed for that*. To this end, both unsupervised (i.e. techniques designed to identify patterns/clusters by examining unlabeled data e.g. Romero et al. 2015, Chen et al., 2018) and supervised techniques (i.e. employing representative labeled/training data which are used by a learning approach that will generate an inferred function mapping the input to its corresponding output e.g. Charou et al., 2019) were examined in the literature to address land cover classification problems, with the latter being investigated in this thesis as well. Specifically, the kNearest Neighbors and Random Forests approaches were investigated in this thesis due to their wide employment in remote sensing applications (Liya and Schulz 2015, Schäfer et al. 2019, Abdi 2019).

2.2.1. kNearest Neighbors Classification

kNearest Neighbors is a supervised, non-parametric, proximity-based, machine learning algorithm (IBM, 2023). The algorithm assigns a class label utilizing plural voting by examining the kNearest samples to the provided input (Figure 1). If k = 1 then the algorithm simply assigns to the input the class label of the nearest sample. Identifying k-values may require extensive experimentation since low k-values may lead to high variance/low bias, and high k-values may lead to lower variance/higher bias. Usually selecting an optimum k-value depends on the input dataset.

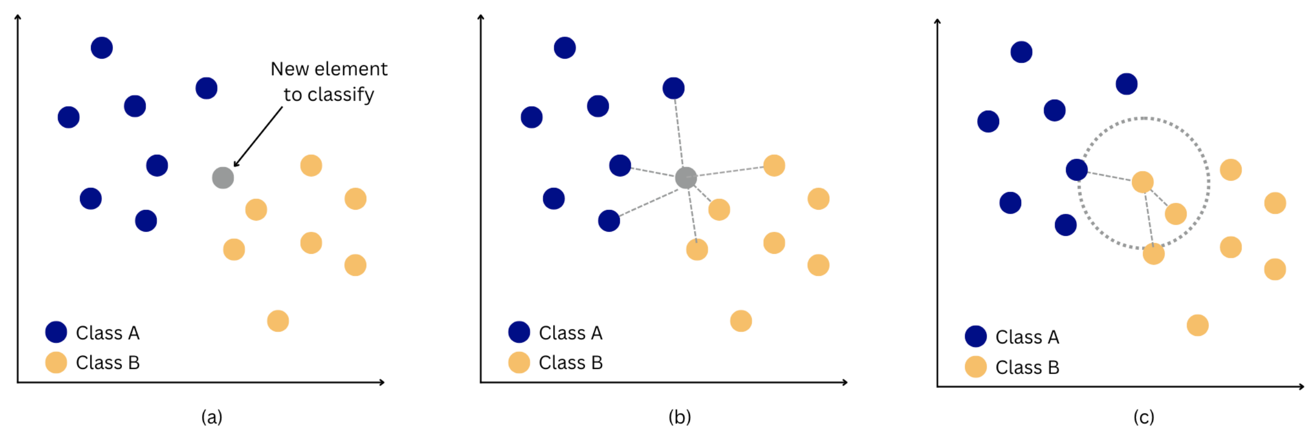


Figure 1: kNearest Neighbor algorithm (k=3). (a) A new unlabeled element enters the algorithm (b) The algorithm calculates the distance between the input element and all other instances in the dataset. (c) The algorithm assigns a class label utilizing plural voting by examining the 3 nearest samples to the provided input

Different metrics were utilized in the literature to compute the distance between the input and the labeled samples, with some of the most widely adopted being:

- *Euclidean Distance*: the most commonly used distance measure, limited to real-valued vectors. It measures a straight line between two points:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad \text{(Equation 1.1)}$$

n: number of vector elements

- *Manhattan (or City-Block) Distance*: It measures the absolute value between two points. It can be conceived as the movement one could do when navigating from one grid point to another

(similar to moving from one city block to another)

$$d(x, y) = \sum_{i=1}^{n} \left| x_i - y_i \right| \qquad \text{(Equation 1.2)}$$

n: number of vector elements

- *Minkowski Distance*: A generalized distance equation which by setting proper values to its *p-parameter* can be specialized in both Euclidean (p=2) and Manhattan (p=1) distances. Different p-values can derive additional distance equations.

$$d(x, y) = \left(\sum_{i=1}^{n} \left| x_i - y_i \right|^p\right)^{\frac{1}{p}} \qquad \text{(Equation 1.3)}$$

n: number of vector elements

2.2.2. Random Forests Classification

Random Forests is an ensemble method (i.e. a method utilizing multiple learning algorithms to improve their predictive performance) that constructs multiple relatively uncorrelated decision trees during training (Breiman, 2001). When predicting the classification output, it assigns the label which is predicted by the most decision trees in the forest.

A decision tree can be conceived as a graph having two node types. A conceptualization is presented in Figure 2.

- *Decision Node*s have multiple branches (usually utilizing a dichotomic approach with two major branches). Based on the outcome of the Decision Node, a certain branch is followed which may lead to another Decision Node or a Leaf Node.
- *Leaf Nodes* are used when a final decision should be reached by the parent Decision Node.

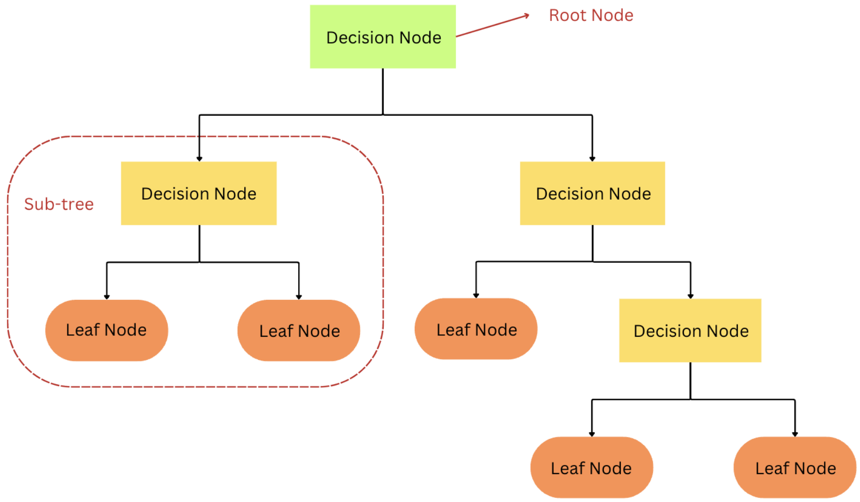


Figure 2: Conceptualization of a Decision Tree.

To automatically train a Classification Decision Tree, usually a greedy divide and conquer approach is used. Assuming a dataset (X, Y - X: samples, Y: labels) having f independent variables (features) describing its properties, a common training approach is to divide the original training set into different subsets (dictated by the input labels) based on dichotomous independent variables (e.g. is ndvi > 0.2). The process can be called recursively to further split the resulting sub-population until the dataset can be split no more, or a certain stopping condition is met. This training approach is called recursive partitioning (Breiman, 1984).

The basic Random Forests training phase can be described with the following pseudocode (B: Number of iterations, n: number of samples)

For b = 1 to B:

- Create an (X_b, Y_b) dataset by uniformly sampling with replacement n samples from the original training dataset,
- Train a Classification Decision Tree with (X_b, Y_b).

To reduce the correlation of the resulting trees, Random Forests may also select prior to the training of the Classification Decision Tree a subset of the features originally provided in the training set (Ho, 2002).

3. Materials and Methods

This chapter presents the materials and datasets used to implement the project, as well as the location and main land use categories of the study area. This is followed by the methodology adopted to produce the results: from the collection and preparation of the data to the training and evaluation of the machine learning model and, finally, the land cover classification.

3.1. Datasets

The data employed in the project include multispectral and thermal imagery, as well as terrain information over the area of study. Regarding the first two dataset types, Landsat 8 and Landsat 9 images were selected for this thesis since they include both multispectral and thermal information in a single acquisition. As for the elevation dataset, ASTER GDEM was employed.

3.1.1. Landsat satellite imagery

The Landsat program started in the early 1970s with the launching of Landsat 1, formerly known as Earth Resources Technology Satellite. It has included nine satellites over its history, of which two are currently operational: Landsat 8, launched on 11 February 2013, and Landsat 9, launched on 27 September 2021. They both feature two sensors; one is the Operational Land Imager (OLI), providing multispectral imagery in the visible, near-infrared (NIR), and shortwave infrared (SWIR) regions of the electromagnetic spectrum. The second is the Thermal Infrared Sensor (TIRS), which generates imagery in the thermal infrared. The spatial resolution of each of these Landsat sensors is illustrated in Table 1.

Table 1: Landsat 8-9 Operational Land Imager (OLI) and Thermal Infrared Sensor (TIRS) (source: https://www.usgs.gov/faqs/what-are-band-designations-landsat-satellites)

Band number	Band name	Wavelength (µm)	Spatial resolution (m)
1	Coastal aerosol	0.43-0.45	30
2	Blue	0.45-0.51	30
3	Green	0.53-0.59	30
4	Red	0.64-0.67	30
5	Near InfraRed (NIR)	0.85-0.88	30
6	SWIR 1	1.57-1.65	30
7	SWIR 2	2.11-2.29	30
8	Panchromatic	0.50-0.68	15
9	Cirrus	1.36-1.38	30
10	Thermal Infrared (TIRS) 1	10.6-11.19	100
11	Thermal Infrared (TIRS) 2	11.50-12.51	100

The Operational Land Imager (OLI) collects data from nine spectral channels, of which only seven correspond to the channels of the previous satellites of Landsat legacy. Two new channels have been added to Landsat 8 and 9, one for water quality assessment (Band 1) and one to improve the detection of fine clouds in the upper atmosphere (thunderclouds, Band 9). The Thermal InfraRed Sensor (TIRS) measures ground temperature and the data provided are also used in applications related to water management applications. The technology available is used in two channels in the thermal infrared, making it possible to separate the temperatures of the Earth's surface temperatures from those of the atmosphere and thus providing better estimates of temperature measurements compared to the previous Landsat receivers, which have a thermal channel.

3.1.2. Elevation dataset

Information related to the elevation profile of the area will be retrieved with the ASTER GDEM product. According to USGS (2023), "the Terra Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) Global Digital Elevation Model (GDEM) Version 3 (ASTGTM) provides a global digital elevation model (DEM) of land areas on Earth at a spatial resolution of 1 arc second", or 30x30 meters.

It was generated using ASTER Level-1A scenes that were acquired between March 2000 and November 2013 (NASA et al., 2018). With this information, second-level terrain derivative datasets can be further generated, e.g. slope, aspect, and/or topographic positioning index. Even though several studies suggest that the specific product is outperformed in terms of accuracy compared to other similar datasets (Han et al. 2021, Yao et al. 2020, Rana et al., 2019), ASTER GDEM has low sensitivity to land cover and specifically better quality in forest areas than that in the cropland/ grassland/bare land on a flat surface (Satgé et al. 2018).

3.2. Study area

The study area of the thesis is located in Greece over the Ionian Islands and their adjacent mainland, which includes part of Epirus, and Sterea Ellada regions (Figure 3). This area covers approximately 323 sq. km. and consists of both island and continental areas of the Greek Peninsula.

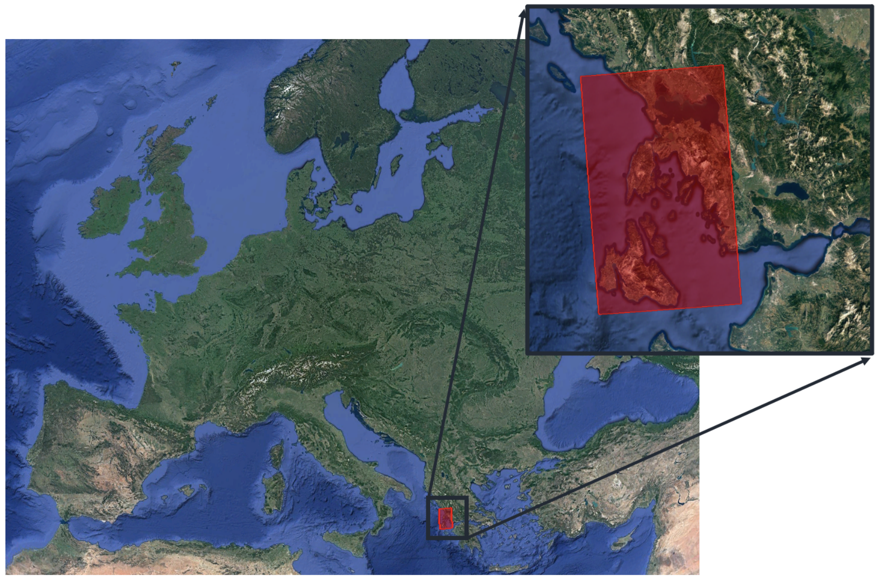


Figure 3: Area of interest and broader region covering 76.44 x 126.78 km, or approximately 323 sq.km. (Source: Google Aerial)

The CORINE Land Cover product, which stands for 'Coordination of information on the environment', provides a pan-European land cover and land use dataset covering 44 classes. It is a three-level hierarchical classification scheme that classifies homogeneous landscape patterns, which have more than 75% of the properties of a specified nomenclature class (Copernicus Land Monitoring Service, 2023). The minimum cartographic unit equals 25 ha - and approximately equal to 277 Landsat pixels (30x30m) - and it has a geometric accuracy better than 100 m, which is the product's spatial resolution. Updated products are released every six years, with the most recent to be made for 2018 (Copernicus Land Monitoring Service, 2023). The major land cover categories (Level 1 classes) are:

- Artificial surfaces
- Agricultural areas
- Forest and semi-natural areas
- Wetlands
- Water bodies

With respect to the land coverage of the area, a screenshot of the CORINE Land Cover product of 2018 shows that the existing categories are related mainly to agriculture (yellowish colors in Figure 4A) and open vegetation areas and forests (greenish colors in the figure). Urban areas (red color in the figure) and water features, such as rivers, estuaries, and inland marshes (light blue colors in the Figure 4A) have an adequate presence over the whole study area.

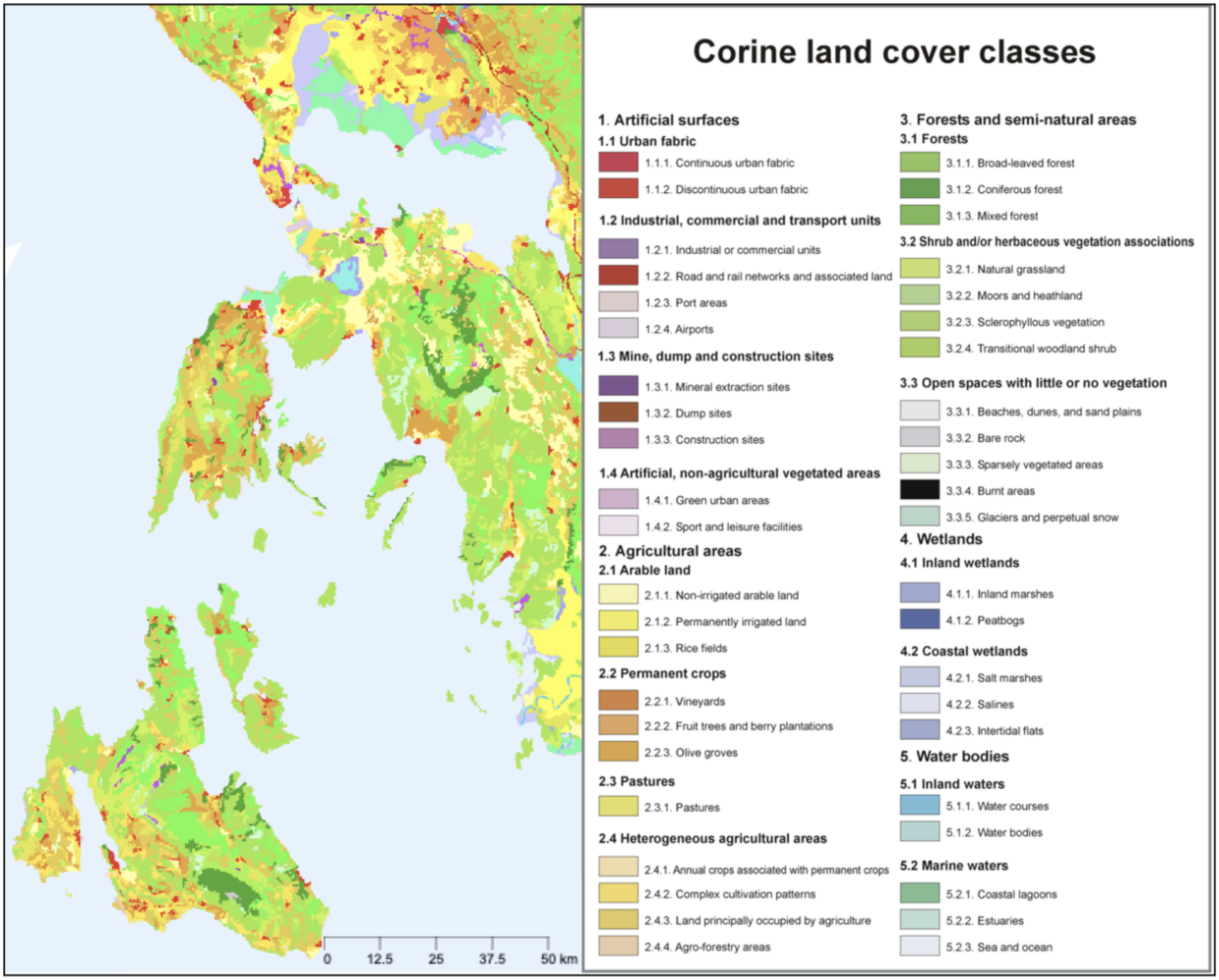


Figure 4A: CORINE Land Cover 2018 of the study area.

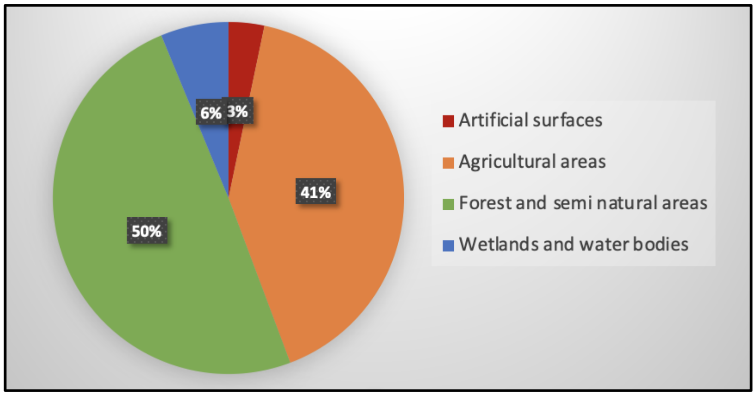


Figure 4B: Level-1 land cover classes and coverage percentages of the study area as extracted from CORINE Land Cover 2018.

After an analysis of the study area, and according to the CORINE Land Cover dataset of 2018, half of it is covered by forests and semi-natural areas, whereas 41% of the land cover is croplands and agricultural regions. Only 6% and 3% of the total land area is covered by wetlands/water bodies and artificial surfaces respectively. Thus, it includes all of the main land cover class features in a sufficient quantity for this study's purposes.

As to the climate conditions of the area, according to Köppen climate classification, it has a Mediterranean climate characterized by temperate dry, and hot summers (Csa) dominantly. An indicative profile of these climates is presented in the following table, which refers to the monthly means of temperature, precipitation, humidity, and sunshine hours of Corfu Island, as measured by the Hellenic National Meteorological Service and NOAA (Table 1). It can be observed that it rains throughout the entire year, with higher rainfall measurements between November and December.

Climate data for Corfu [hide]													
Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Year
Record high °C (°F)	20.5 (68.9)	22.4 (72.3)	26.0 (78.8)	28.0 (82.4)	33.8 (92.8)	38.0 (100.4)	43.0 (109.4)	40.0 (104.0)	37.4 (99.3)	31.0 (87.8)	25.0 (77.0)	22.0 (71.6)	43.0 (109.4)
Average high °C (°F)	13.9 (57.0)	14.2 (57.6)	16.0 (60.8)	19.0 (66.2)	23.8 (74.8)	28.0 (82.4)	30.9 (87.6)	31.3 (88.3)	27.6 (81.7)	23.2 (73.8)	18.7 (65.7)	15.3 (59.5)	21.8 (71.2)
Daily mean °C (°F)	9.7 (49.5)	10.3 (50.5)	12.0 (53.6)	14.9 (58.8)	19.6 (67.3)	23.9 (75.0)	26.4 (79.5)	26.3 (79.3)	22.7 (72.9)	18.4 (65.1)	14.3 (57.7)	11.1 (52.0)	17.5 (63.5)
Average low °C (°F)	5.1 (41.2)	5.7 (42.3)	6.8 (44.2)	9.2 (48.6)	12.9 (55.2)	16.4 (61.5)	18.4 (65.1)	18.8 (65.8)	16.5 (61.7)	13.4 (56.1)	9.9 (49.8)	6.8 (44.2)	11.7 (53.1)
Record low °C (°F)	−4.5 (23.9)	−4.2 (24.4)	−4.4 (24.1)	0.0 (32.0)	4.6 (40.3)	8.7 (47.7)	10.0 (50.0)	11.3 (52.3)	7.2 (45.0)	2.8 (37.0)	−2.2 (28.0)	−2.0 (28.4)	−4.5 (23.9)
Average rainfall mm (inches)	136.6 (5.38)	124.6 (4.91)	98.1 (3.86)	66.7 (2.63)	37.0 (1.46)	14.1 (0.56)	9.2 (0.36)	19.0 (0.75)	81.3 (3.20)	137.7 (5.42)	187.4 (7.38)	185.6 (7.31)	1,097.3 (43.20)
Average rainy days	16.1	14.6	14.5	12.9	8.0	4.9	2.3	3.4	7.0	11.8	15.7	17.5	128.7
Average relative humidity (%)	75.4	74.3	73.4	72.8	69.5	63.4	60.0	62.2	70.4	74.6	77.5	77.2	70.7
Mean monthly sunshine hours	117.7	116.8	116.0	206.5	276.8	324.2	364.5	332.8	257.1	188.9	133.5	110.9	2,545.7

Figure 5: Monthly means of temperature (°C, °F), precipitation (mm, inches), humidity (%), and sunshine hours of Corfu Island (Source: Hellenic National Meteorological Service and NOAA)

As shown in Figure 6, the elevation profile of the area contains a varying topography including both steep and plain areas, with the highest point at 1592 meters according to ASTER GDEM.

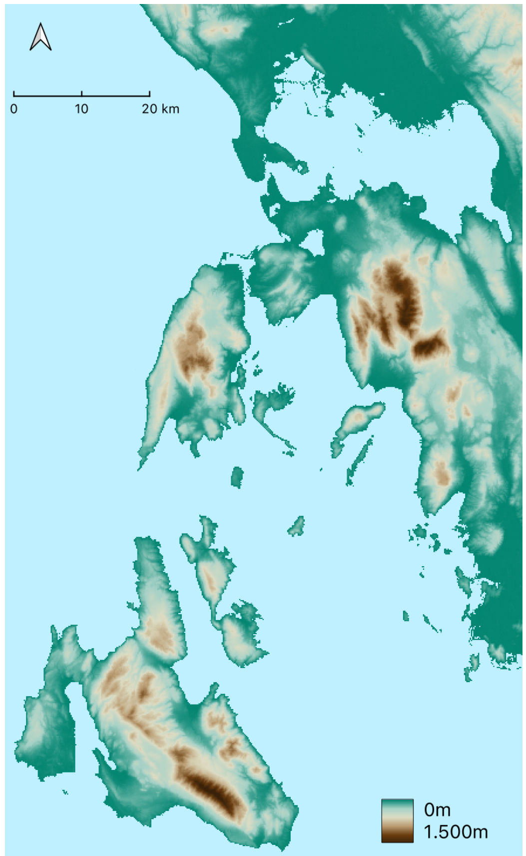


Figure 6: Elevation map of the study area.

3.3. Methodology

The methodology for the implementation of the thesis project includes several steps. These can be summarized as follows:

- Data collection
- Data pre-processing
- Data preparation and Machine Learning (ML) model training
- Application of the selected ML method for the land-cover classification
- Presentation and evaluation of results

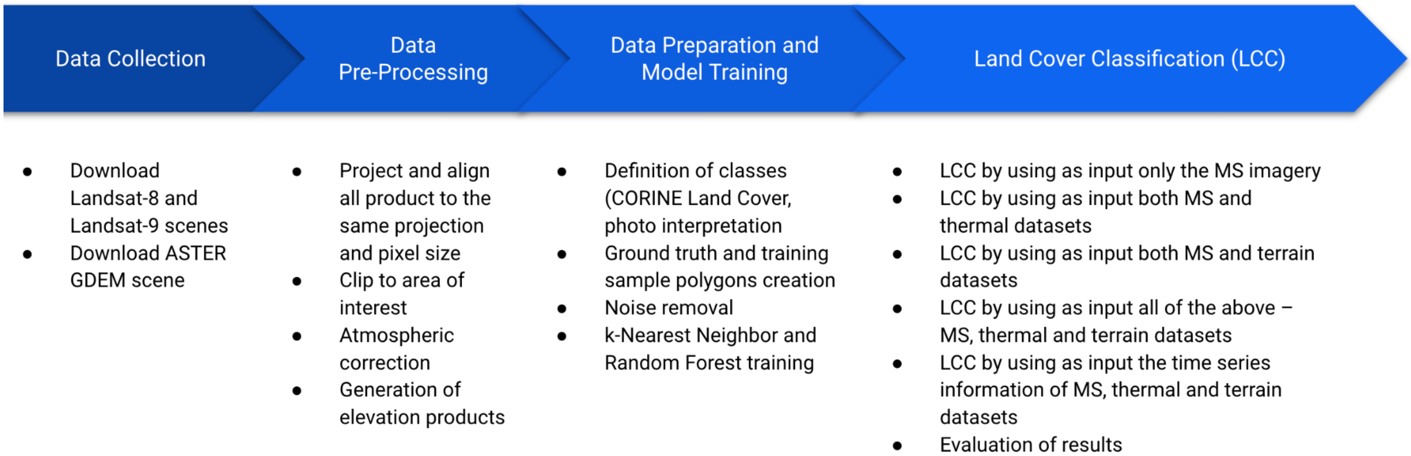


Figure 7: Methodology steps.

3.3.1. Data collection

The first step of the method was the dataset collection. The sources from which the datasets were retrieved are shown in the following table:

Table 2: Input datasets and sources.

	Product(s)	Source
Multispectral and Thermal imagery	Landsat 8 and Landsat 9	USGS EarthExplorer (https://earthexplorer.usgs.gov/)
Elevation data	ASTER GDEM V3	NASA JPL ASTER (https://asterweb.jpl.nasa.gov/gdem.asp)

An essential parameter for the satellite imagery acquisition taken into consideration was, apart from the bounding box coordinates of the AOI, the extent of cloud coverage in each scene. A percentage of less than 10% of cloud occurrence was selected to be defined as a restriction for imagery downloading. In this way, a minimization of the noise reduction, and, thus, the dataset pre-processing overall performance was achieved.

Table 3: Spatial resolution and temporal coverage of input data.

Sensor	Spatial resolution	Revisit time	Temporal coverage
Landsat 8 OLI/TIRS	30m/100m	16 days	Since February 2013
Landsat 9 OLI/TIRS	30m/100m	16 days	Since September 2021
ASTER GDEM V3	30m	-	-

A total number of 88 available Landsat scenes that cover the area of interest were downloaded, out of which 76 were captured from Landsat 8 sensor and 12 from Landsat 9 (Table 4).

Table 4: Available Landsat scenes with less than 10% cloud occurrence.

Year	**Available datasets (<10% cloud coverage)**	
	Landsat 8	**Landsat 9**
2013	9	-
2014	8	-
2015	8	-
2016	8	-
2017	8	-
2018	6	-
2019	7	-
2020	7	-
2021	7	-
2022	8	12
TOTAL	***88***	

Table 5: Landsat scene acquisition dates and corresponding season used in this study.

	Winter (11 scenes)			**Spring** (12 scenes)			**Summer** (39 scenes)			**Autumn** (26 scenes)		
	Dec	**Jan**	**Feb**	**Mar**	**Apr**	**May**	**Jun**	**Jul**	**Aug**	**Sep**	**Oct**	**Nov**
2013	8				28		15	1	2, 18	3, 19	21	
2014		9	10	30				4, 20	5		8	9
2015			13			4, 20		7, 23	24		27	12
2016					4, 20		7, 23	9	10, 26		13	
2017		1			23		10, 26	12, 28		14	16	
2018			5			12		15, 31		1, 17		
2019							16	2, 18	3, 19	4	22	
2020							18	4, 20	21	6	24	25
2021						4		7	8, 24	25	27	12
2022	25	15, 23, 31	24		5	15, 31		2, 10, 18, 26	19, 27	4, 12, 20	22, 30	7
TOTAL	***2***	***5***	***4***	***1***	***5***	***6***	***7***	***19***	***13***	***11***	***10***	***5***

3.3.2. Data pre-processing

The data pre-processing consisted of clipping the downloaded datasets to the study area and their alignment at the pixel level, the atmospheric correction of the multispectral satellite imagery, and the generation of two elevation products using terrain analysis, having as input the ASTER GDEM elevation dataset.

Clipping and Alignment

To make use of the elevation model and the Landsat 8 and Landsat 9 scenes, initially, the datasets must be in the same coordinate system and projection, and cover the same area.

As a first step, all the imagery was reprojected to the WGS 84 / UTM zone 34N (EPSG:32634) using the Warp tool in the QGIS software. This step is mandatory in order to prevent any miscalculations between the rasters during the classification process. The next step was the clipping of all imagery to the boundaries of the region of interest using the *Superimpose* application of OrfeoToolbox, and the downscaling of the thermal bands from 100 meters to 30 meters of spatial resolution. *Superimpose* (Orfeo ToolBox, 2023) performs the projection of an image into the geometry of another one, having as a result the first image obtains the same spatial resolution and occupies the same physical space as the reference image.

The final images have a size of 2548 x 4226 pixels and a spatial resolution of 30 meters, which corresponds to an area of interest covering 76.44 x 126.78 km, or approximately 323 sq.km.

Atmospheric correction

The Landsat bands used in this study are Band 2 (Blue), Band 3 (Green), Band 4 (Red), Band 5 (NIR), Band 6 (SWIR1), Band 7 (SWIR2), Band 10 (TIRS1), and Band 11 (TIRS2). Since the downloaded Landsat scenes are Level-1 products, further processing is needed to convert the digital values of the imagery into atmospherically corrected ground reflectance values. For the two thermal channels, the image pre-processing procedure is followed with the conversion of the reflectance values to brightness temperature values. The atmospheric correction equations and the process followed are described below.

1. **Conversion from Digital Numbers (DN) to Top of Atmosphere (TOA) values**

Landsat Level-1 data are converted to TOA spectral radiance using the radiance rescaling factors in the metadata file of each scene using the following equation:

$$L_\lambda = M_L Q_{cal} + A_L \quad \text{(Equation 2.1)}$$

where Lλ is the spectral irradiance (Watts/(m^2 *srad*μm)), ML and AL coefficients derived from the metadata file and Qcal is the DN of the pixel.

2. Conversion to Brightness Temperature (BT) values

The TOA radiation values are then reduced to temperature values (brightness temperature) in degrees on the Kelvin scale, based on the following equation:

$$BT = \frac{K_2}{\ln\left(\frac{K_1}{L_\lambda}+1\right)} \quad \text{(Equation 2.2)}$$

where BT is the brightness temperature (°K), Lλ is the spectral irradiance (Watts/(m^2 * srad * μm)) and K1, K2 coefficients derived from the file metadata file. To convert Kelvin degrees to degrees Celsius, the equation is used:

$$BT_C = BT_K - 273.15 \quad \text{(Equation 2.3)}$$

Calculation of elevation products

Two elevation products that can be generated from a Digital Elevation Model and were used in the present study are slope and topographic positioning index.

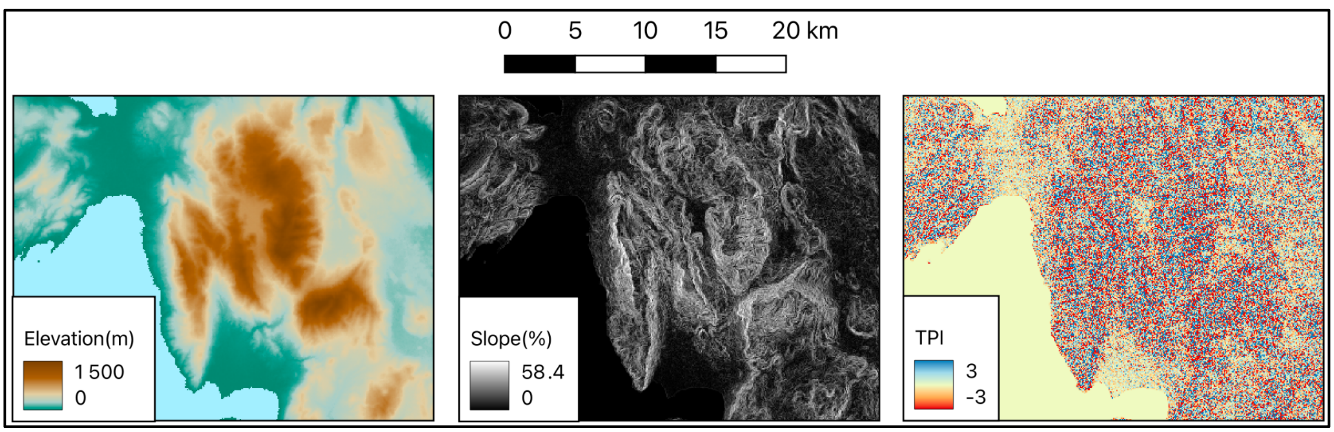
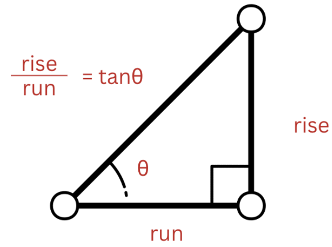


Figure 8: Digital elevation model, Slope, and Topographic position index of a sub-region over the area of interest.

Slope is defined as the rate of change of elevation for each cell of a Digital Elevation Model, or the steepness of the surface. The slope value is calculated by measuring the angle between the topographic surface and the referenced datum. Both planar and geodesic computations are performed using a 3 by 3 cell neighborhood (moving window). The formula that transforms elevation to the slope is the following:



$$Slope = arctan\left(\frac{rise}{run}\right) \times 100$$ Equation 2.4

Topographic Positioning Index (TPI) measures the difference between elevation at the central point (z_0) and the average elevation ($\underline{z}$) around it within a predetermined radius (R) (Wilson and Gallant, 2000, Weiss, 2001):

$$\text{TPI} = z_0 - \underline{z}$$ (Equation 2.5)

$$\underline{z} = 1\frac{1}{n_R}\sum_{i \in R} z_i$$ (Equation 2.6)

Positive TPI values indicate that the central point is located higher than its average surroundings, thus are indicative of ridges or hilltops. Negative values indicate a position lower than the average, indicating valley features in the topography of the area. TPI values close to 0 indicate straight slopes and/or plain regions (Knitter et al. 2019).

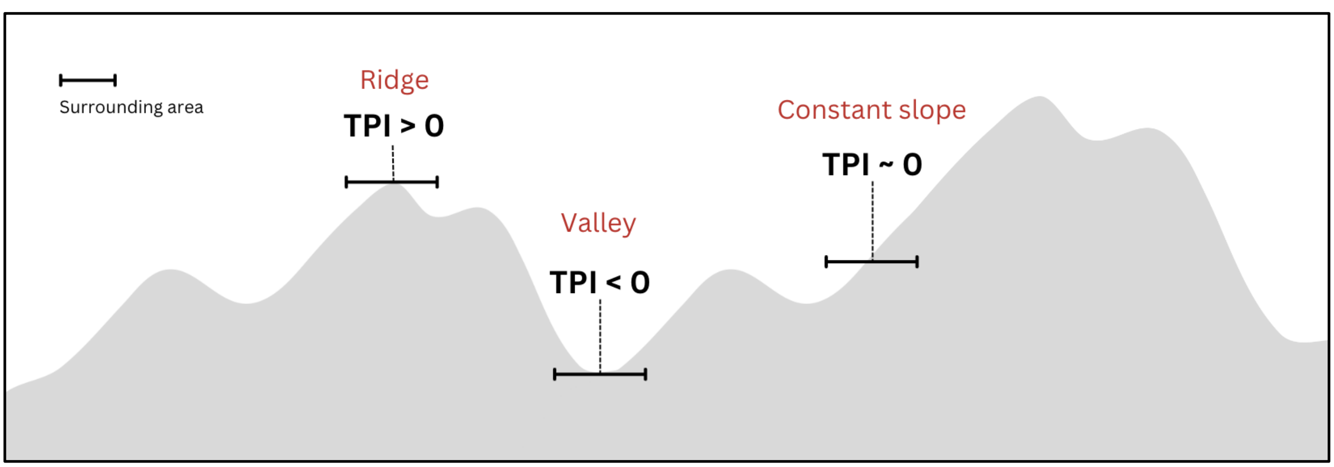


Figure 9: Topographic position index values.

3.3.3. Data preparation

The next step of the methodology is the preparation for the land cover classification. The classification will be supervised, which means that the classes are specific, known, and work as input to the classification algorithm. The required steps include the selection of the classes, the manual collection of training and ground truth samples, the data cleaning, and the generation of time series that will be used as input in the classification process. Each step is described in the next sections.

Selection of classes

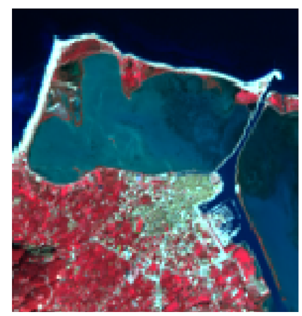
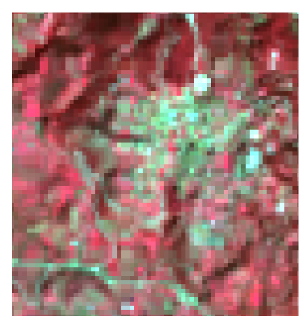
According to Chapter 2.2, where the review of the area of interest was implemented, there are five main classes that cover the majority of the region and these are manmade (or artificial) surfaces, agricultural areas, forests, semi-natural areas, wetlands, and water bodies. After a more detailed photointerpretation of the area using a Landsat scene acquired in May 2018 - in order to match the CORINE Land Cover product for the year 2018, eight classes were identified and selected for the classification process: 1. Artificial, 2. Bare Soil, 3. Cropland, 4. Dense Forest, 5. Grassland, 6. Low-density Urban, 7. Low Sparse Vegetation, and 8. Water (Table 6). Photointerpretation method and examples of each class are presented in Table 7.

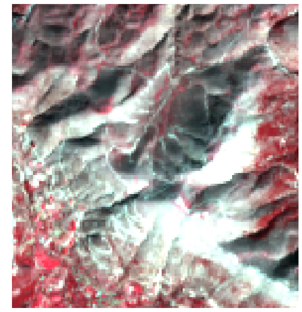
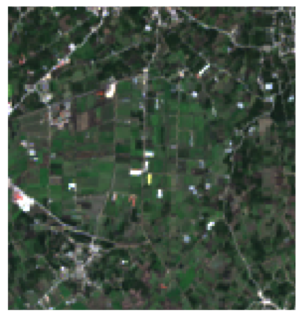
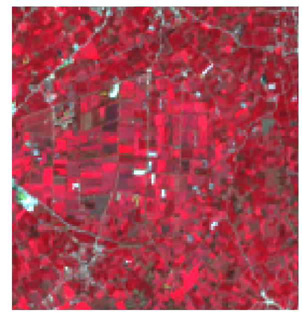
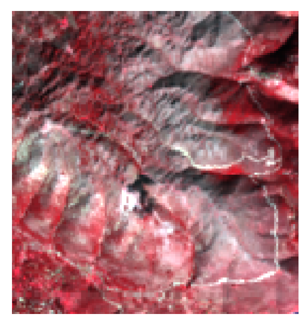
Table 6: The description and correspondence to CORINE Land Cover (CLC) classification of each class used in this study.

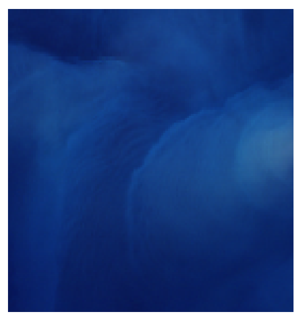
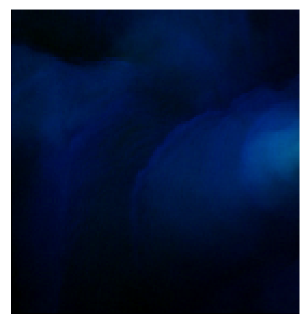
Class name	**Description**	**Correspondence/ Similarity to CLC classification**
Artificial	Man-made surfaces. It includes urban regions with no green areas, industrial sites, transportation networks, commercial areas, recreational spaces, landfills, and mining areas.	1.1.1 1.2 1.3.1
Low- density Urban	Urban areas with sparse buildings interrupted by vegetation or bare soil.	1.1.2
Bare Soil	Uncovered land with no vegetation or growth (soil, rock, sand, etc)	3.3.1 3.3.2
Cropland	Seasonal agricultural areas.	2
Low Sparse Vegetation	Areas with limited and scattered plant cover.	3.2.3 3.2.4 3.3.3
Grassland	Open areas dominated by grasses and lack significant tree or woody vegetation.	2.3 3.2.1
Dense Forest	Areas with thick, abundant tree cover.	3.1
Water	Bodies of water, such as sea, lakes, rivers, and ponds.	5

In the context of this study, it is important to acknowledge that not all CORINE Land Cover classes could be directly correlated with the chosen classes. This is primarily due to two reasons: either certain classes were found to be absent in the study area based on the initial analysis, or they were present but in a less dominant capacity.

Table 7: The photointerpretation method and the appearance on a true color composite TCC (RGB-432) and a false color positive FCC (RGB-543) over the study area of each class used in this study. For the photointerpretation, a Landsat-8 true color and false color image acquired on 12/05/2018 was used, and a very high spatial resolution basemap was exploited for cross-reference.

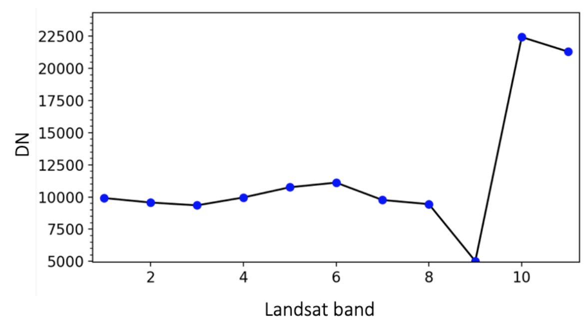
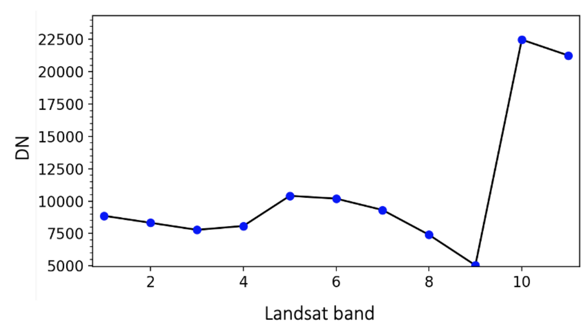
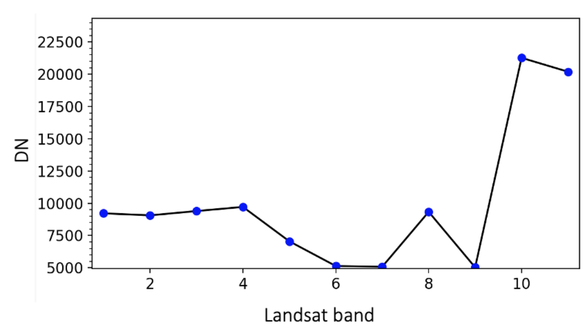
Class	**Photointerpretation**		**Example in study area**	
			TCC (RGB-432)	**FCC (RGB-543**
Artificial	Color in TCC	Mixed white, gray and brown shades with minimal to no vegetation (green areas)		
	Color in FCC	Mixed white, green and yellow shades with minimal to no vegetation (red areas)		
	Other characteristics	Dense, structured patterns.		
Low-density Urban	Color in TCC	Mixed white and brown shades with more visible vegetation (green areas)		
	Color in FCC	Mixed white, green and yellow shades with more visible vegetation (red areas)		
	Other characteristics	Buildings may be scattered rather than forming continuous blocks or clusters. Also slightly more greenery compared to high-density artificial zones.		
Bare Soil	Color in TCC	Shades of brown, white or gray, with no vegetation (green).		

	Color in FCC	Shades of brown, white or gray, with no vegetation (red).		
	Other characteristics	Smoother texture compared to vegetated areas. May include open spaces, construction sites, or agricultural fields.		
Cropland	Color in TCC	Patches of varying colors (green, white or brown) depending on the crop type and growth stage.		
	Color in FCC	Patches of varying colors (red, white or brown) depending on the crop type and growth stage.		
	Other characteristics	Well-defined rectangular or geometric shapes often with less natural vegetation than the surrounding landscape. Presence of human-made features such as irrigation systems, farm structures, or roads within or adjacent to the cropland areas.		
Low Sparse Vegetation	Color in TCC	Pale green and brown tones.		
	Color in FCC	Pale red and brown tones.		
	Other characteristics	Irregular shapes of limited vegetative cover, with a higher proportion of bare ground or soil compared to regions with denser vegetation.		
Grassland	Open areas with dominant green hues in a true color inage, indicating grass cover and minimal tree or	Vibrant, uniform green color.		
		Vibrant, uniform red color.		

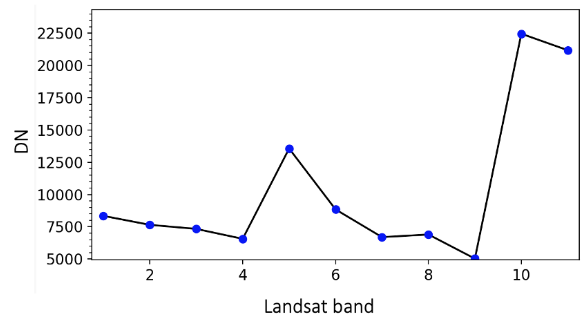
	building presence	Even and conitnuous surfaces, usually surrounded by forests.		
Dense Forest	Color in TCC	Rich and dark green colors.		
	Color in FCC	Rich and vibrant red colors.		
	Other characteristics	Continuous and solid surfaces.		
Water	Color in TCC	Deep and light blue shades		
	Color in FCC	Blue or black areas		
	Other characteristics	Uniform and relatively featureless compared to land surfaces, with distinct boundaries (coastline, river banks, etc).		

For the classes' photointerpretation, the Landsat scene of 12 May 2018 was used, in order to match the reference year of the CORINE Land Cover dataset. The Landsat true color composite (RGB-432) and false color composite (RGB-543) were created and used for visualization purposes when digitizing the samples, and a very high spatial resolution basemap was exploited as ancillary data. In this study, Bing Maps product was used as the ancillary cross-reference dataset thanks to its easy integration into the QGIS. The selection of these classes was evaluated through their spectral signatures. As a spectral signature of a surface, it is defined as the amount of radiation reflectance of the specific surface in the electromagnetic spectrum (ESA-Eduspace, 2009). The spectral signature, as well as the assigned color of each class, are shown in Table 8.

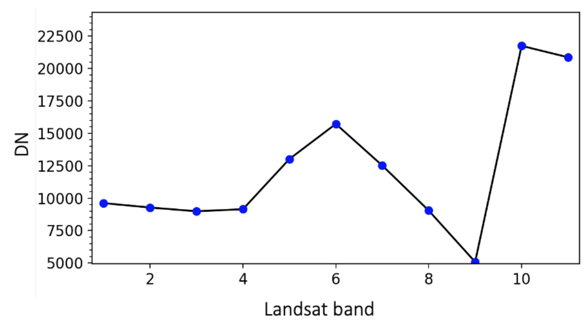
Table 8: The spectral signature of each class as extracted from random pixels that represent the correspondent classes over the study area. The x-axis represents the Landsat bands as described in Table 1, while the y-axis displays the Digital Numbers (DN) or pixel values of each band.

Class	**Spectral Signature**
Artificial	
Low-density Urban	
Bare Soil	

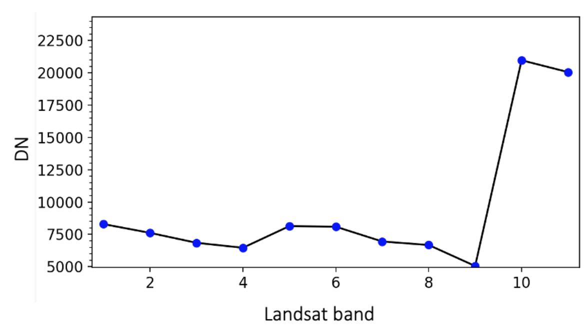
Cropland



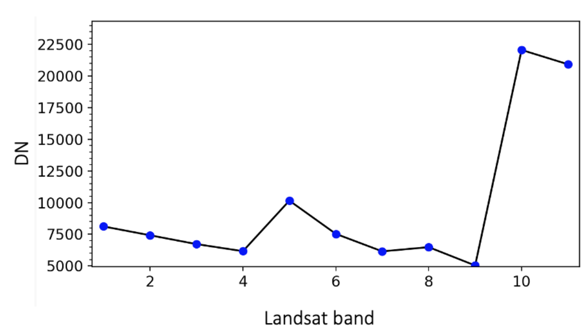
Low Sparse Vegetation

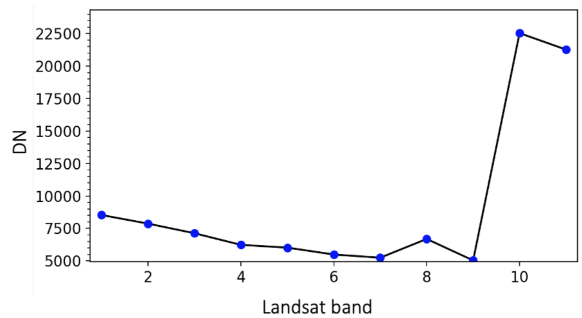


Grassland



Dense Forest



Water	

As observed in Table 8, ‘Water’, ‘Bare soil’, ‘Low/sparse vegetation’ and ‘Artificial’ classes have very distinct spectral signatures, which means that they can be easily distinguished by an algorithm. The spectral signatures that could be confused due to their pattern similarity are between ‘Artificial’ and ‘Low Density Urban’, ‘Dense Forest’ and ‘Cropland’, and ‘Grassland’ and ‘Low density urban’. The variations in all three cases that differentiate the respective signatures are:

- between ‘Artificial’ and ‘Low Density Urban’, ‘Low Density Urban; has lower pixel values in the optical (Red - Green - Blue) and Near Infrared bands (Bands 1, 2, 3, 4)
- between ‘Dense Forest’ and ‘Cropland’, ‘Cropland’ has a higher value in Band 5 (Near Infrared),
- between ‘Grassland’ and ‘Low density urban’, ‘Low density urban’ class has higher reflectance in the Near Infrared and SWIR bands (Bands 5, 6, 7).

Collection of training and ground truth sample polygons

For each class, 23 polygons were digitized manually using the Landsat scene of 12 May 2018, after following the photointerpretation guidelines presented in Table 7. Ancillary data, such as the CORINE Land Cover 2018 dataset and Bing Maps, were also used. In order to match the epoch of the ancillary data, the reference Landsat scene used for the sample generation was acquired within 2018.

Some examples of the selected samples are shown in Figure 10.

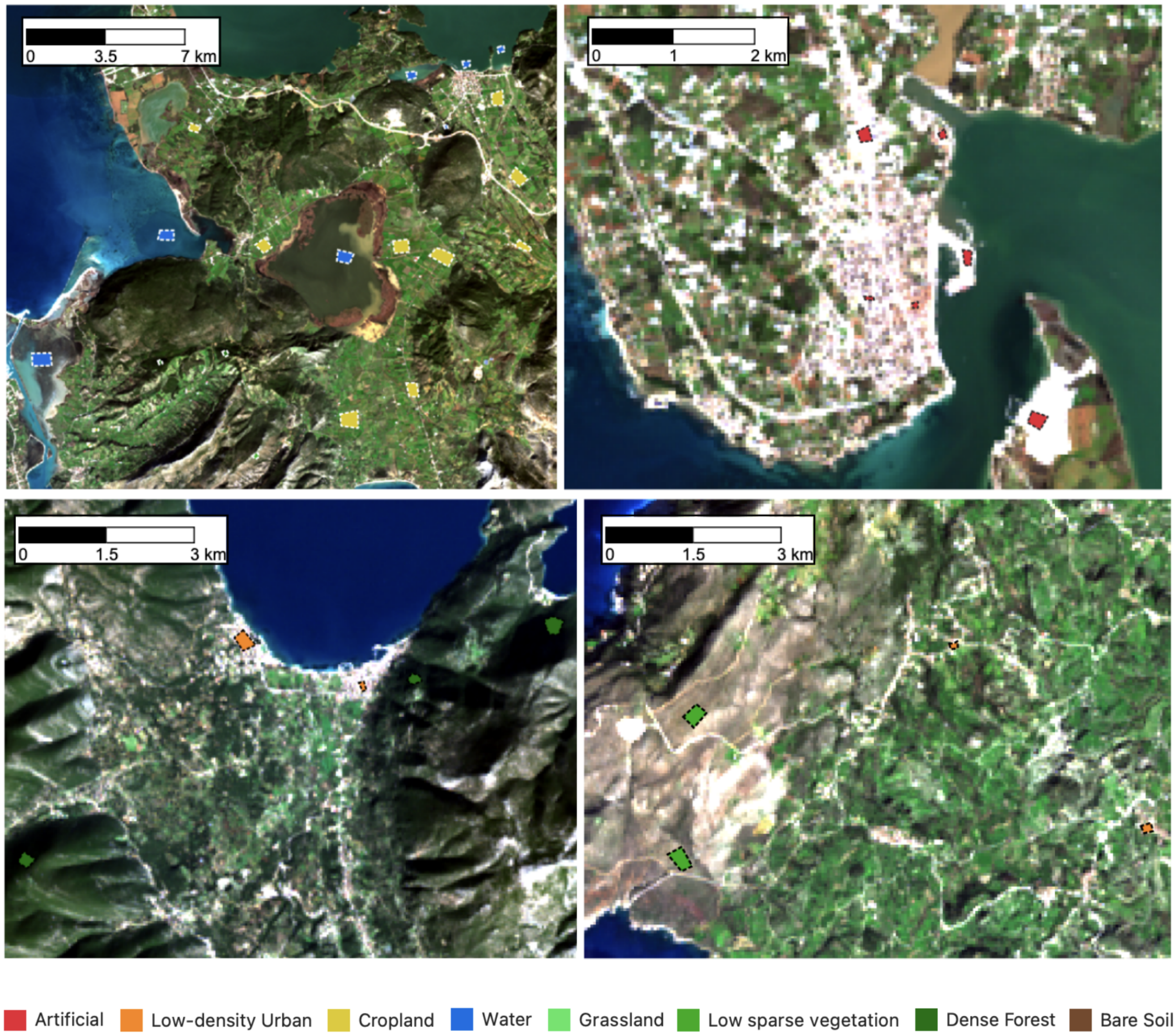


Figure 10: Sample selection of the different land cover classes from Landsat 8 scene acquired on 12/05/2018 (RGB-432).

Noise removal

Even though an atmospheric correction was performed, noisy pixels existed in several images, mainly due to cloud coverage and shadows. To eliminate these noisy pixels (e.g. cloud-covered) it was assumed that the polygon values of each class follow a normal distribution and that the abrupt noise was caused only due to clouds and shadows. Thus, to cut off the 1% of each edge, the mean value and the standard deviation were computed and the values in the range [xmean - 3*std, xmean+3*std] were preserved.

Generation of training and ground truth pixel samples

To perform supervised pixel-based classification for each (a) Landsat imagery (88 scenes x 8 bands) and (b) elevation products (elevation, slope, TPI) all the corresponding pixel values from the polygon samples and for each class were extracted from the data. A total number of 1,895,168 pixel-samples were calculated for all classes and per scene. From these and for each class, 80% was randomly selected to be the training samples and 20% to be used as ground truth samples for evaluation. The total and per class number of training and ground truth pixel samples that were collected in this study are presented in Table 9.

Table 9: The total and per class training and ground truth samples used in the study.

Land Cover Class	**Training (Pixels)**	**Ground Truth (Pixels)**
Artificial	14,925	3,731
Bare Soil	38,790	9,698
Cropland	421,274	105,318
Dense Forest	83,706	20,926
Grassland	31,187	7,797
Low Density Urban	25,274	6,318
Low/Sparse Vegetation	120,314	30,078
Water	780,666	195,166
TOTAL	*1,516,136*	*379,032*

Generation of the time-series dataset

The next step was the organization of both training and ground truth samples in a time sequence for the classification using time-series of all datasets. To do this, all samples were sorted by class, date (Table 5), and data contents. Data contents included the multispectral bands (Band 2, Band 3, Band 4, Band 5, Band 6, Band 7), and the thermal bands (Band 10, Band 11) of the available Landsat 8 and Landsat 9 scenes. This procedure generated a dataset containing the pixel values for each band throughout time, labelled by class. Elevation product values remained stable throughout time, since they were generated from a single dataset, so they were added at a second stage in the time-series. The pixel time series values for all available dates are presented for each band used in this study in Annex B.

3.3.4. Machine Learning model training and evaluation

Training is the process of passing to a Machine Learning (ML) model the prepared dataset in order to learn patterns and relationships from the data and make predictions that are better than it would have been without training. In this study, two different ML algorithms were used: Random Forests and k-Nearest Neighbor. As shown in previous chapter, these ML algorithms are two of the mostly used in the literature, and with good classification results. For the implementation of the ML training, the Python programming language was used, and the scikit-learn Python package. The entire Python script can be found in Annex C.

Nine different models with different inputs for each algorithm were created to produce land cover classifications (LCCs), according to the five classification approaches of this thesis project (Table 10). The parameters used for each classifier were the defaults, since the scope of this study was to compare the land cover classification performance according to the different data inputs. For the kNearest Neighbors, the selected (by default) number of neighbors was set to 5, and the number of trees for the Random Forests classifier was set to 100. Since the default maximum depth of the Random Forests' trees was set to 'None', the number of features in each tree was equal to the total number of training samples as calculated in Table 9. All the scikit-learn library's default parameters of the kNearest Neighbors and Random Forests classifiers used in this study are presented in Table 11.

Table 10: The nine classification Machine Learning models created in the study.

Land Cover Classification	Input	Algorithm
1A	Multispectral	kNearest Neighbor
1B	Multispectral	Random Forests
2A	Multispectral and thermal	kNearest Neighbor
2B	Multispectral and thermal	Random Forests
3A	Multispectral and terrain	kNearest Neighbor
3B	Multispectral and terrain	Random Forests
4A	Multispectral, thermal and terrain	kNearest Neighbor
4B	Multispectral, thermal and terrain	Random Forests
5	Time series of multispectral, thermal and terrain	Random Forests

Table 11: The scikit-learn library's default parameters of the kNearest Neighbors and Random Forests classifiers used in the study.

Machine Learning Classifier	Scikit-learn Module (v1.3.2)	Main Parameters (default)
kNearest Neighbors	sklearn.neighbors.KNeighborsClassifier	● Number of neighbors (n) = 5 ● Weight function used in prediction: uniform. All points in each neighborhood are weighted equally. ● Algorithm used to compute the nearest neighbors: 'auto'. It attempts to decide the most appropriate algorithm based on the values passed to fit method. ● Metric to use for distance computation: standard Euclidean distance
Random Forests	sklearn.ensemble.RandomForestClassifier	● The number of trees in the forest (n) = 100 ● Function to measure the quality of a split: Gini ● Maximum depth of the tree: None ● Minimum number of samples required to split an

		internal node: 2 ● Minimum number of samples required to be at a leaf node: 1 ● Number of features to consider when looking for the best split: sqrt

It should be noted that due to the computational complexity of the model creation of the time series dataset, kNearest Neighbors was slow and its model was not calculated. Thus, the ML model for the time series dataset was generated only using the Random Forests classifier.

After the training of the ML models, the evaluation of the models' performance took place using the ground truth data as validation. This was performed by calculating the confusion matrices per model, whose values indicate how well each model would work on new data.

3.3.5. Land cover classification

The application of the trained Random Forests and kNearest Neighbor models to the entire dataset was the next step of the method. Nine different classification results were generated, one for each LCC approach using the corresponding inputs. The final results are presented in Chapter 3.

3.3.6. Accuracy assessment

The post-classification accuracy assessment has been considered as the most vital part of validating the LULC maps produced (Manandhar 2009, Hurskainen 2019). In this study, the performance of each classification experiment was calculated with respect to the selected study area and compared with the rest of the classification outputs, both per class and as an overall classification score using User Accuracy, Producer Accuracy, and Kappa coefficient metrics. The equations used for the calculation of each metric are the following:

Overall Accuracy: $$acc = \frac{\Sigma A}{N} \quad \text{(Equation 2.7)}$$

User Accuracy: $$acc = \frac{A}{C} \quad \text{(Equation 2.8)}$$

Producer Accuracy: $$acc = \frac{A}{B} \quad \text{(Equation 2.9)}$$

Kappa (Classification): $$acc = \frac{Nd - q}{N^2 - q} \quad \text{(Equation 2.10)}$$

Kappa (per class): $$acc = \frac{Ndi - qi}{NBi - qi} \quad \text{(Equation 2.11)}$$

where:

- A, refers to the correctly mapped sampling points for each class (diagonal of confusion matrix),
- B, refers to the total number of ground truth points for each class,
- C, refers to the total number of map data points for each class,
- N, refers to the total number of sampling points,
- d, refers to the sum of correctly mapped points,
- q, refers to the sum of the products between B and C for each class.

User accuracy refers to the correctly mapped sampling points per category, whereas producer accuracy refers to the correctly interpreted ground truth points per category. The Kappa Coefficient ranges from -1 to 1. A value of 0 indicates that the classification is no better than a random classification. A negative number indicates the classification is significantly worse than random. A value close to 1 indicates that the classification is significantly better than random (Humboldt State University, 2019).

4. Results

In this chapter, the classification results of the entire area of interest and the calculated confusion matrices of each classification model are presented, as well as the land cover classification results over three different sub-regions of the entire area of interest. These areas have been selected carefully to include all the classes, so that the differences in the results of the classification approaches and the different algorithms used in this study are visible. The accuracy assessment metrics, both per class and as an overall score, are also presented in this chapter.

The designations used in the sub-chapters for the different classification approaches in terms of inputs are presented in Table 12.

Table 12: Designations of the different classification approaches used in the presentation of results.

Classification Approach	**Input**
1	Multispectral (MS) imagery
2	MS and thermal information
3	MS and terrain information
4	MS, thermal and terrain information
5	Time series information of all of the above - MS, thermal and terrain

4.1. Classification results

The classification results generated by the five different classification approaches and the two machine learning algorithms (kNearest Neighbor and Random Forests) are shown in Figure 11. Overall, it is observed that water areas were classified very satisfactorily accurately in all cases. Furthermore, vegetation classes (*'Dense Forest'*, *'Low/Sparse Vegetation'*, *'Grassland'*), as well as *'Bare Soil'*, seem to be strongly affected by the topography of the area. This is visible in classifications 3A, 3B, 4A, 4B, and 5 of Figure 11, where terrain information was used as input in the classification process. The same is applied also for the category *'Cropland'*; it is better interpreted and delineated in the classifications where terrain information is incorporated. In terms of evaluating the algorithms, the main observation is that *'Bare Soil'* surfaces are best classified with the kNearest Neighbor. The kNearest Neighbor algorithm also seems to overestimate artificial surfaces (classification results 3A and 4A) compared to the Random Forests algorithm.

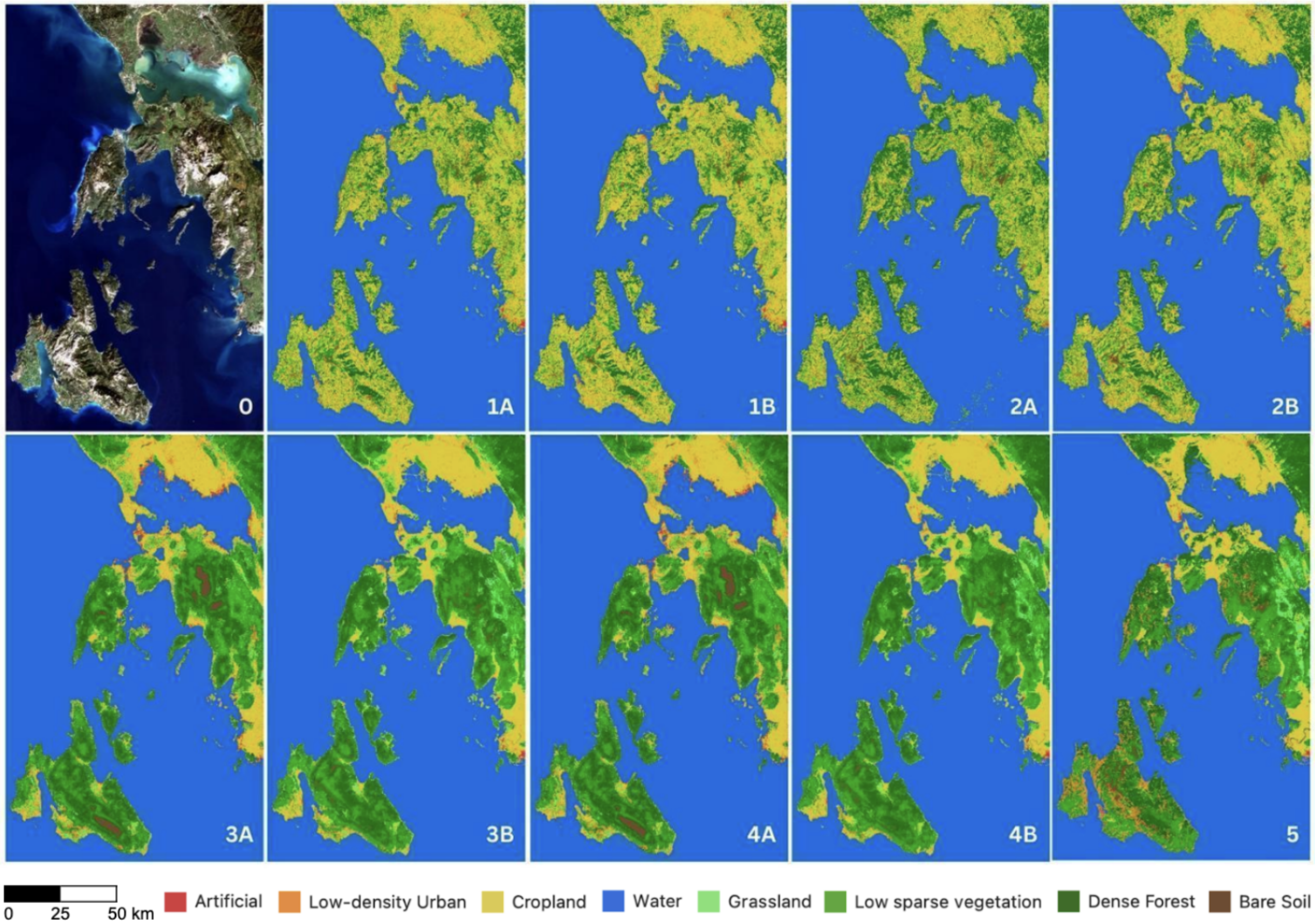


Figure 11: Overall qualitative inspection of the land cover classification results having as reference the original Landsat-8 true color image (acquisition date: 12/05/2018, RGB = 432) of the full study area using kNearest Neighbor (A) and Random Forests (B) algorithms. (0) True color composite, (1) Classification result using only multispectral imagery (2) Classification result using MS and thermal information, (3) Classification result using MS and terrain information, (4) Classification result using MS, thermal, and terrain information, (5) Classification result using the time series of MS, thermal, and terrain information with Random Forests.

4.2. Confusion matrices

This section showcases the results of all methods developed on the test dataset. The evaluation of each model was implemented using as a reference the ground truth sample dataset per class, which is independent of the training dataset. The confusion matrices - one for the kNearest Neighbor and one for the Random Forests algorithm, as well as the classification accuracies (User Accuracy, Producer Accuracy, Kappa, Overall) achieved by each method are presented in Tables 13A and 13B.

For the classifications where kNearest Neighbor machine learning algorithm is used (Table 13A), *'Artificial'* is generally confused with *'Cropland'* when MS, thermal, and terrain information are incorporated as inputs in the algorithm (CA4), whereas there is a confusion at lower levels mainly with *'Low Density Urban'*. '*Bare Soil'* class shows a higher confusion level with *'Cropland'* and *'Low/Sparse Vegetation'*, mainly when only MS and the combination of MS and thermal data are used as inputs in the classification. The 'Cropland' class is mainly mixed with the *'Low/Sparse Vegetation'*, while a confusion also exists with the classes *'Grassland'*, *'Bare Soil'*, and *'Dense Forest'*. However, when terrain information is used, these confusions are relatively eliminated. The highest confusion for the

'Dense Forest' class is with 'Cropland' in classifications where input data were MS and both MS and thermal information. As for the *'Grassland'* category, this is the only class that is highly confused with the *'Cropland'* class in CA1 and CA2. The remainder of classification approaches, in which terrain information is used as input, show good results. For the *'Low Density Urban'* class, it is observed that it is largely confused with *'Cropland'* in classification approaches that incorporate thermal information (CA2 and CA4). *'Low/Sparse Vegetation'* is mostly confused with the *'Cropland'* category and at lower levels with *'Bare Soil'* in CA1 and CA2. CA3 shows the lowest confusion results for this class, while in CA4 *'Low/Sparse Vegetation'* is slightly confused with all classes apart from *'Artificial'* and *'Water'*. *'Water'* shows the lowest confusion results compared to the rest of the classes. Only in CA2, where the input datasets are both MS and thermal information, a generic confusion is observed, which nevertheless exists at low levels.

In general, for the classifications performed using the kNearest Neighbor algorithm, and especially in CA1 and CA2, a confusion between the majority of classes and the class 'Cropland' is observed. Furthermore, the classification approach with the lowest confusion level between the different classes is CA3, where the algorithm inputs were the MS and the terrain products.

Table 13A: Confusion matrices and classification accuracies of the different Classification Approaches (CA) for kNearest Neighbor algorithm.

CA	kNearest Neighbor (n=5)
1	

	Class	Ground Truth Data							
		Artificial	Bare Soil	Cropland	Dense Forest	Grassland	Low Density Urban	Low/Sparse Vegetation	Water
Map Data	Artificial	**3373**	30	70	9	5	283	18	7
	Bare Soil	71	**6303**	1197	77	22	70	1908	6
	Cropland	158	870	**96603**	789	1276	636	4773	26
	Dense Forest	8	85	868	**19884**	16	23	218	5
	Grassland	9	67	4574	87	**2494**	5	427	20
	Low Density Urban	244	103	994	49	2	**4784**	92	2
	Low/Sparse Vegetation	20	1565	7239	231	177	43	**20585**	6
	Water	14	15	20	13	2	3	13	**195639**

Overall Accuracy = 0.92
Kappa = 0.88

Class	User Accuracy	Producer Accuracy	Kappa
Artificial	0.89	0.87	0.86
Bare Soil	0.65	0.70	1.03
Cropland	0.92	0.87	1.00
Dense Forest	0.94	0.94	1.00
Grassland	0.32	0.62	1.06
Low Density Urban	0.76	0.82	1.03
Low/Sparse Vegetation	0.69	0.73	1.01
Water	1.00	1.00	1.00

2

Class		Ground Truth Data							
		Artificial	Bare Soil	Cropland	Dense Forest	Grassland	Low Density Urban	Low/Sparse Vegetation	Water
Map Data	Artificial	**2641**	253	318	14	6	349	199	15
	Bare Soil	206	**4936**	1818	99	84	215	2235	61
	Cropland	147	1179	**94877**	1131	1267	500	5870	160
	Dense Forest	33	76	1206	**19187**	30	35	175	365
	Grassland	8	164	4600	140	**2089**	21	621	40
	Low Density Urban	198	355	2084	66	16	**2597**	930	24
	Low/Sparse Vegetation	71	1578	9353	350	350	357	**17723**	84
	Water	21	22	114	131	15	22	40	**195354**

Overall Accuracy = 0.89
Kappa = 0.84

Class	User Accuracy	Producer Accuracy	Kappa
Artificial	0.70	0.79	0.79
Bare Soil	0.51	0.58	1.05
Cropland	0.90	0.83	1.00
Dense Forest	0.91	0.91	1.00
Grassland	0.27	0.54	1.07
Low Density Urban	0.41	0.63	1.07
Low/Sparse Vegetation	0.59	0.64	1.01
Water	1.00	1.00	1.00

3

Class		Ground Truth Data							
		Artificial	Bare Soil	Cropland	Dense Forest	Grassland	Low Density Urban	Low/Sparse Vegetation	Water
Map Data	Artificial	**3781**	0	14	0	0	0	0	0
	Bare Soil	0	**9654**	1034	48	7	0	0	0
	Cropland	8	0	**105120**	317	441	3	0	0
	Dense Forest	0	1	0	**21104**	3	0	2	0
	Grassland	0	0	0	0	**7682**	0	1	0
	Low Density Urban	0	0	6	0	0	**6264**	0	0
	Low/Sparse Vegetation	0	0	0	0	0	1	**29865**	0
	Water	0	0	0	0	0	0	7	**195719**

Overall Accuracy = 0.97
Kappa = 0.99

Class	User Accuracy	Producer Accuracy	Kappa
Artificial	1.00	1.00	1.00
Bare Soil	0.90	1.00	1.00
Cropland	0.99	0.99	1.00
Dense Forest	1.00	0.98	1.00
Grassland	1.00	0.94	1.01
Low Density Urban	1.00	1.00	1.00
Low/Sparse Vegetation	1.00	1.00	1.00
Water	1.00	1.00	1.00

4

Class		*Ground Truth Data*							
		Artificial	Bare Soil	Cropland	Dense Forest	Grassland	Low Density Urban	Low/Sparse Vegetation	Water
Map Data	Artificial	**2192**	0	1559	0	10	34	0	0
	Bare Soil	0	**9332**	0	178	1	3	140	0
	Cropland	178	0	**104258**	0	213	403	62	17
	Dense Forest	0	94	0	**20731**	8	35	239	0
	Grassland	3	1	363	27	**7085**	18	186	0
	Low Density Urban	34	0	1536	48	18	**4521**	105	8
	Low/Sparse Vegetation	0	126	145	461	213	127	**28794**	0
	Water	0	0	111	0	0	2	0	**195606**

Overall Accuracy = 0.96
Kappa = 0.97

Class	User Accuracy	Producer Accuracy	Kappa
Artificial	0.58	0.91	0.91
Bare Soil	0.97	0.98	1.00
Cropland	0.99	0.97	1.00
Dense Forest	0.98	0.97	1.00
Grassland	0.92	0.94	1.01
Low Density Urban	0.72	0.88	1.02
Low/Sparse Vegetation	0.96	0.98	1.00
Water	1.00	1.00	1.00

The classification results generated from the Random Forests algorithm (Table 13B) show a similar trend to those from the kNearest Neighbor algorithm, but with differences to some categories. *'Artificial'* shows low confusion results in all classifications, with a slight confusion with *'Low Density Urban'* in CA1 and CA2. The *'Cropland'* class is mainly mixed with the *'Low/Sparse Vegetation'*, again in CA1 and CA2. A confusion also exists with the classes *'Grassland'*, *'Low Density Urban'* and *'Dense Forest'*. However, as with kNearest Neighbor, when terrain information is used, these confusions are relatively eliminated. *'Grassland'*, on the other hand, is largely confused with *'Cropland'* in the classification approach where only MS data are used (CA1). When thermal information is also used for the classification (CA2), this confusion is almost halved, but still is at high levels. As in kNearest Neighbor classifications, the classification approaches where terrain information is used as input show good results for this category. *'Low/Sparse Vegetation'* is, again, mostly confused with the *'Cropland'* category and at lower levels with *'Bare Soil'* in CA1 and CA2. Finally, *'Water' class* shows the lowest confusion results compared to the rest of the classes.

Table 13B: Confusion matrices and classification accuracies of the different Classification Approaches (CA) for Random Forests ML algorithms.

CA	Random Forests (n = 100)
1	see below
2	see below

CA 1

	Class	Ground Truth Data							
		Artificial	Bare Soil	Cropland	Dense Forest	Grassland	Low Density Urban	Low/Sparse Vegetation	Water
Map Data	**Artificial**	**3363**	24	75	5	1	296	13	18
	Bare Soil	78	**5786**	1365	67	17	54	2277	10
	Cropland	115	269	**99867**	580	707	433	3116	44
	Dense Forest	0	44	876	**19903**	8	17	247	12
	Grassland	18	17	4717	78	**2457**	6	362	28
	Low Density Urban	212	45	1125	41	4	**4730**	109	4
	Low/Sparse Vegetation	17	1021	7901	164	81	28	**20643**	11
	Water	2	6	23	12	3	0	4	**195669**

Overall Accuracy = 0.92
Kappa = 0.89

Class	User Accuracy	Producer Accuracy	Kappa
Artificial	0.89	0.88	0.88
Bare Soil	0.60	0.80	1.02
Cropland	0.95	0.86	1.00
Dense Forest	0.94	0.95	1.00
Grassland	0.32	0.75	1.04
Low Density Urban	0.75	0.85	1.03
Low/Sparse Vegetation	0.69	0.77	1.01
Water	1.00	1.00	1.00

CA 2

	Class	Ground Truth Data							
		Artificial	Bare Soil	Cropland	Dense Forest	Grassland	Low Density Urban	Low/Sparse Vegetation	Water
Map Data	**Artificial**	**3376**	17	83	5	2	285	7	20
	Bare Soil	206	**6625**	1034	48	7	55	1799	8
	Cropland	147	176	**101778**	317	441	393	1899	14
	Dense Forest	33	25	625	**20301**	3	9	133	5
	Grassland	8	14	3743	50	**3627**	7	232	3
	Low Density Urban	198	34	1104	28	1	**4795**	111	4
	Low/Sparse Vegetation	9	741	5554	122	56	19	**23358**	7
	Water	0	3	6	10	0	0	7	**195693**

Overall Accuracy = 0.94
Kappa = 0.92

Class	User Accuracy	Producer Accuracy	Kappa
Artificial	0.89	0.85	0.85
Bare Soil	0.68	0.87	1.01
Cropland	0.97	0.89	1.00
Dense Forest	0.96	0.97	1.00
Grassland	0.47	0.88	1.02
Low Density Urban	0.76	0.86	1.03
Low/Sparse Vegetation	0.78	0.85	1.01
Water	1.00	1.00	1.00

3

	Class	Ground Truth Data							
		Artificial	Bare Soil	Cropland	Dense Forest	Grassland	Low Density Urban	Low/Sparse Vegetation	Water
Map Data	**Artificial**	**3723**	0	61	0	1	9	0	1
	Bare Soil	0	**9411**	0	60	0	0	183	0
	Cropland	30	0	**104941**	1	111	30	17	1
	Dense Forest	0	22	0	**20978**	1	8	98	0
	Grassland	0	1	181	9	**7445**	6	104	0
	Low Density Urban	10	0	181	3	2	**6047**	27	0
	Low/Sparse Vegetation	0	151	54	99	36	12	**29514**	0
	Water	0	0	0	0	0	0	0	**195719**

Overall Accuracy = 0.98

Kappa = 0.99

Class	User Accuracy	Producer Accuracy	Kappa
Artificial	0.98	0.99	0.99
Bare Soil	0.97	0.98	1.00
Cropland	1.00	1.00	1.00
Dense Forest	0.99	0.99	1.00
Grassland	0.96	0.98	1.00
Low Density Urban	0.96	0.99	1.00
Low/Sparse Vegetation	0.99	0.99	1.00
Water	1.00	1.00	1.00

4

	Class	Ground Truth Data							
		Artificial	Bare Soil	Cropland	Dense Forest	Grassland	Low Density Urban	Low/Sparse Vegetation	Water
Map Data	**Artificial**	**3702**	0	72	0	1	20	0	0
	Bare Soil	0	**9414**	0	47	0	0	193	0
	Cropland	38	0	**104931**	1	99	49	13	0
	Dense Forest	0	14	0	**21017**	0	7	69	0
	Grassland	0	1	145	8	**7379**	6	144	0
	Low Density Urban	24	1	224	3	1	**5987**	30	0
	Low/Sparse Vegetation	0	138	41	57	41	13	**29576**	0
	Water	0	0	0	0	0	0	0	**195719**

Overall Accuracy = 0.98

Kappa = 0.99

Class	User Accuracy	Producer Accuracy	Kappa
Artificial	0.98	0.98	0.98
Bare Soil	0.98	0.98	1.00
Cropland	1.00	1.00	1.00
Dense Forest	1.00	0.99	1.00
Grassland	0.96	0.98	1.00
Low Density Urban	0.95	0.98	1.00
Low/Sparse Vegetation	0.99	0.99	1.00
Water	1.00	1.00	1.00

5

Map Data \ Ground Truth Data (Class)	Artificial	Bare Soil	Cropland	Dense Forest	Grassland	Low Density Urban	Low/Sparse Vegetation	Water
Artificial	**40**	0	0	0	0	2	0	0
Bare Soil	0	**111**	0	0	0	0	0	0
Cropland	1	0	**1150**	0	0	0	0	0
Dense Forest	0	0	0	**237**	0	0	0	0
Grassland	0	0	0	0	**98**	0	0	0
Low Density Urban	0	0	1	0	0	**64**	0	0
Low/Sparse Vegetation	0	0	0	0	0	0	**350**	0
Water	0	0	0	0	0	0	0	**2254**

Overall Accuracy = 0.98
Kappa = 1.00

Class	User Accuracy	Producer Accuracy	Kappa
Artificial	0.95	0.98	0.98
Bare Soil	1.00	1.00	1.00
Cropland	1.00	1.00	1.01
Dense Forest	1.00	1.00	1.00
Grassland	1.00	1.00	1.00
Low Density Urban	0.98	0.97	0.97
Low/Sparse Vegetation	1.00	1.00	1.00
Water	1.00	1.00	1.00

4.3. Classification results

This section includes the results of each classification as derived from each approach. For the presentation of these results, three sub-regions of the entire study area were selected for which it was judged that they encompass all of the land use categories studied in this paper, and from which representative conclusions can be drawn. These sub-areas are shown in Figures 12A, 12B, and 12C.

The classification approaches with the **highest overall accuracy score** (OA) are:

- Random Forests, using as input multispectral bands and terrain products (OA = 98%)
- Random Forests, using as input multispectral bands, thermal infrared imagery, and terrain products (OA = 98%)
- Random Forests, using as input the time-series of all datasets (OA = 98%)

The classification approaches with the **lowest overall accuracy score** (OA) are:

- kNearest Neighbor, using as input multispectral and thermal infrared bands (OA = 89%)
- kNearest Neighbor, using as input multispectral bands (OA = 92%)
- Random Forests, using as input multispectral bands (OA = 92%)

The kappa coefficient, as already mentioned, indicates the correctness of the points referring to the minimum statistical correctness. The algorithms with the **highest overall kappa score**, concerning the data inputs, are the following:

- Random Forests, using as input the time-series dataset (Kappa = 1)

- Random Forests, using as input multispectral bands, thermal infrared imagery, and terrain products (Kappa = 0.99)
- Random Forests, using as input multispectral bands and terrain products (Kappa = 0.99)
- kNearest Neighbor, using as input multispectral bands and terrain products (Kappa = 0.99)

The algorithms with the lowest overall kappa score, concerning the data inputs, are the:

- kNearest Neighbor, using as input multispectral bands and thermal infrared imagery (Kappa = 0.84)
- kNearest Neighbor, using as input multispectral bands (Kappa = 0.88)
- Random Forests, using as input multispectral bands (Kappa = 0.89)

Quantitatively, the best classification results achieved in this study were generated from Random Forests, using as input the time-series dataset, with overall accuracy equal to 98% and kappa coefficient equal to 1.00. This means that 98% of the evaluation points were correctly mapped, with a percentage of the map 100% better than the map that would have been produced by chance. On the contrary, the combination of algorithm and classification approach with the lowest performance was the kNearest Neighbor using as input the multispectral and thermal bands. The overall accuracy of this approach is 89% and the kappa coefficient equals 0.84. This means that 89% of the evaluation points were correctly mapped, with a percentage of the map 84% better than the map that would have been produced by chance.

Regarding the per class accuracy, all categories apart from *'Grassland'*, '*Low/Sparse Vegetation*', '*Bare Soil*', and *'Artificial'* performed well in all classification approaches, with more than 70% for both user and producer accuracy. The approaches in which the first three of the above-mentioned classes show the lowest accuracies are those that have as input only the multispectral bands and those that have as input both multispectral and thermal bands for both algorithms. 'Artificial' class has low user accuracy in the kNearest Neighbor algorithm where inputs include all datasets - multispectral bands, thermal bands, terrain products (CA4), whereas water areas (i.e. sea) and 'Dense Forest' were the two classes that performed well in all CAs.

Qualitatively, results extracted from the kNearest Neighbor algorithm appear to have significant speckle noise compared to Random Forests, as it can be observed in Figures 12A, 12B, and 12C. Furthermore, when terrain datasets were used as input, they optimized the results over classes that refer to vegetation and soil (Figures 12A, 12B). On the contrary, artificial areas were overestimated and confused with cropland (Figure 12A). Another finding depicted from the qualitative assessment of the classification results and showed in Figure 12C, is that due to crop seasonality, there was high confusion between cropland and other vegetation classes when using multispectral and thermal infrared bands as inputs in the algorithm (CA2). However, this was something that was fixed when terrain information was additionally used as input (CA4).

Specifically, having as reference the true color scenes, the main observations that can be depicted from the produced land cover areas per class are the following:

- **Artificial**: Man-made surfaces (urban areas, airport, road network, remote infrastructures) have been identified effectively in results generated from Random Forests algorithm. Also, the terrain information lowers the effectiveness of the resulting classification when being used as input together with multispectral band information (CA1).
- **Bare Soil**: This class is mainly observed in mountainous areas over the area of interest, and shows better performance when terrain information is used as input, among others.

- **Cropland**: Croplands have been overestimated in most cases, where there should have been other classes instead (such as artificial, bare soil, and vegetation areas - dense forest, low/sparse vegetation or grassland), and show a better performance with Random Forests. *'Cropland'* class is also classified well when input in the algorithm is the time-series dataset, which takes into consideration the seasonality of cultivation activities.
- **Dense Forest**: In general, this class shows good performance in all approaches. However, shadowed areas, e.g. due to steep slopes, are also classified as '*Dense Forest*'. Additionally, when time-series data are used as input in the classification process (CA5), this class shows an underperformance compared to the reference image (Figure 12C). This may occur because of several factors, such as the canopy seasonality depending on the tree type (deciduous or evergreen trees) that has not been considered as a variable in this study, or other events like tree cutting/wildfires and reforestation activities, which interfere in the time-series of the specific pixel and, thus, its identification as a class with a time-series curve similar to '*Dense Forest*'.
- **Grassland**: As presented in the quantitative analysis and the confusion matrices, *'Grassland'* is a class that is mostly misinterpreted by the algorithms in this study. Better results are shown when terrain information is present, and the classification seems to be the most accurate, when compared to the reference image, in CA4 with Random Forests, where all datasets are used as inputs.
- **Low Density Urban**: '*Low Density Urban*' shows the same findings as *'Artificial'*, since it seems to perform better with Random Forests algorithm, and in CA1 and CA2 cases, where no terrain information is employed as input. The only difference is that it is not interpreted correctly in time-series generated classification results.
- **Low/Sparse Vegetation**: This class shows the almost the same response to the different classification approaches as the *'Grassland'* category, but with better and more compact results generated from Random Forests algorithm.
- **Water**: Water features that correspond to sea have been delineated and classified well enough in all sub-regions. Those that represent other water areas, for example rivers, seem to be sensitive to classifications that incorporate MS and terrain information (CA3).

Most of the aforementioned issues can be justified due to the similarities observed in the spectral signatures of the employed classes (e.g. '*Bare Soil*' and *'Artificial'*, or *'Cropland'* and *'Low/Sparse Vegetation'*).

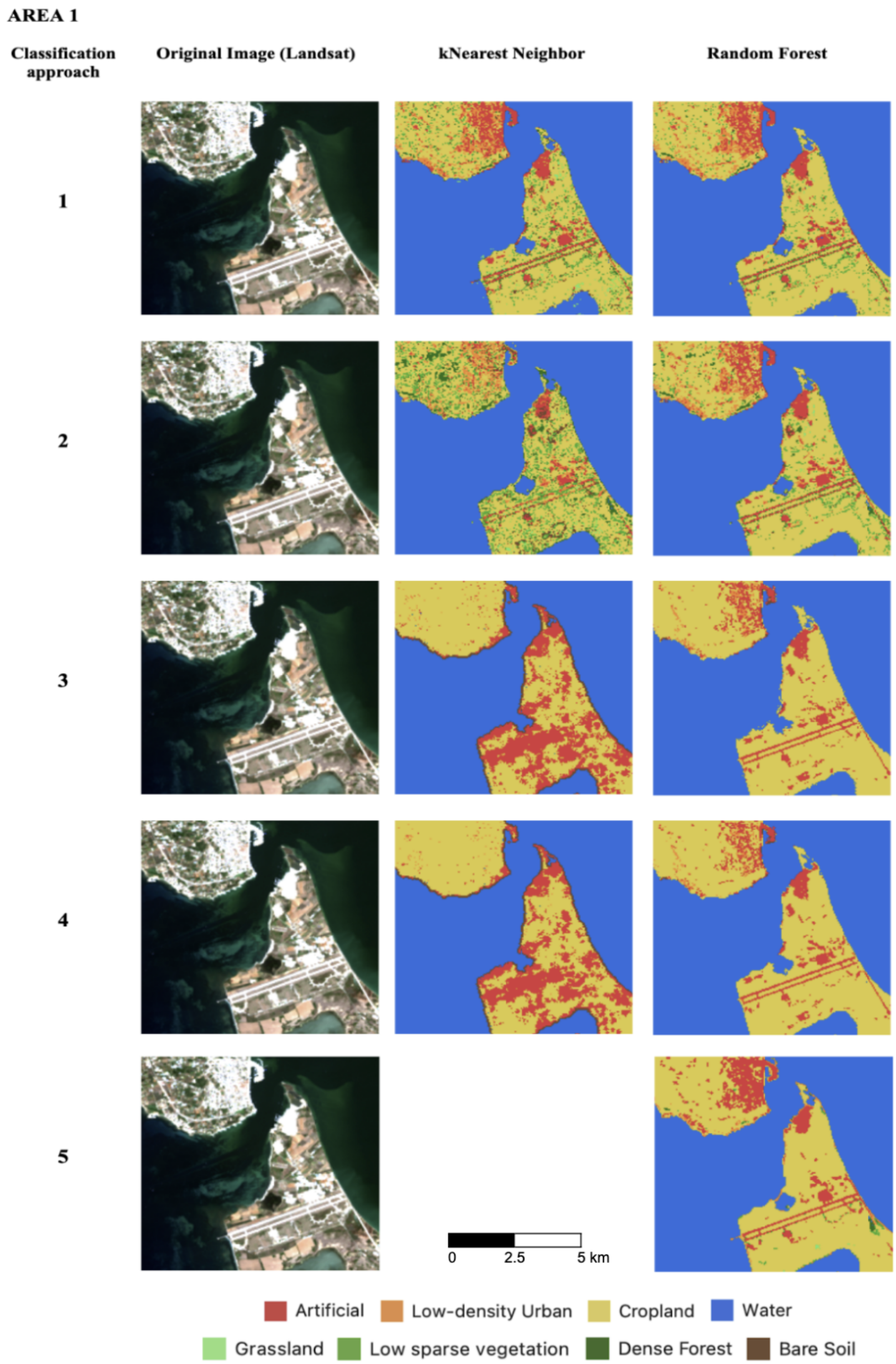


Figure 12A: Original Landsat-8 true color image (acquisition date: 12/05/2018, RGB = 432) and land cover classification results using kNearest Neighbor and Random Forests algorithms for sub-region 1.

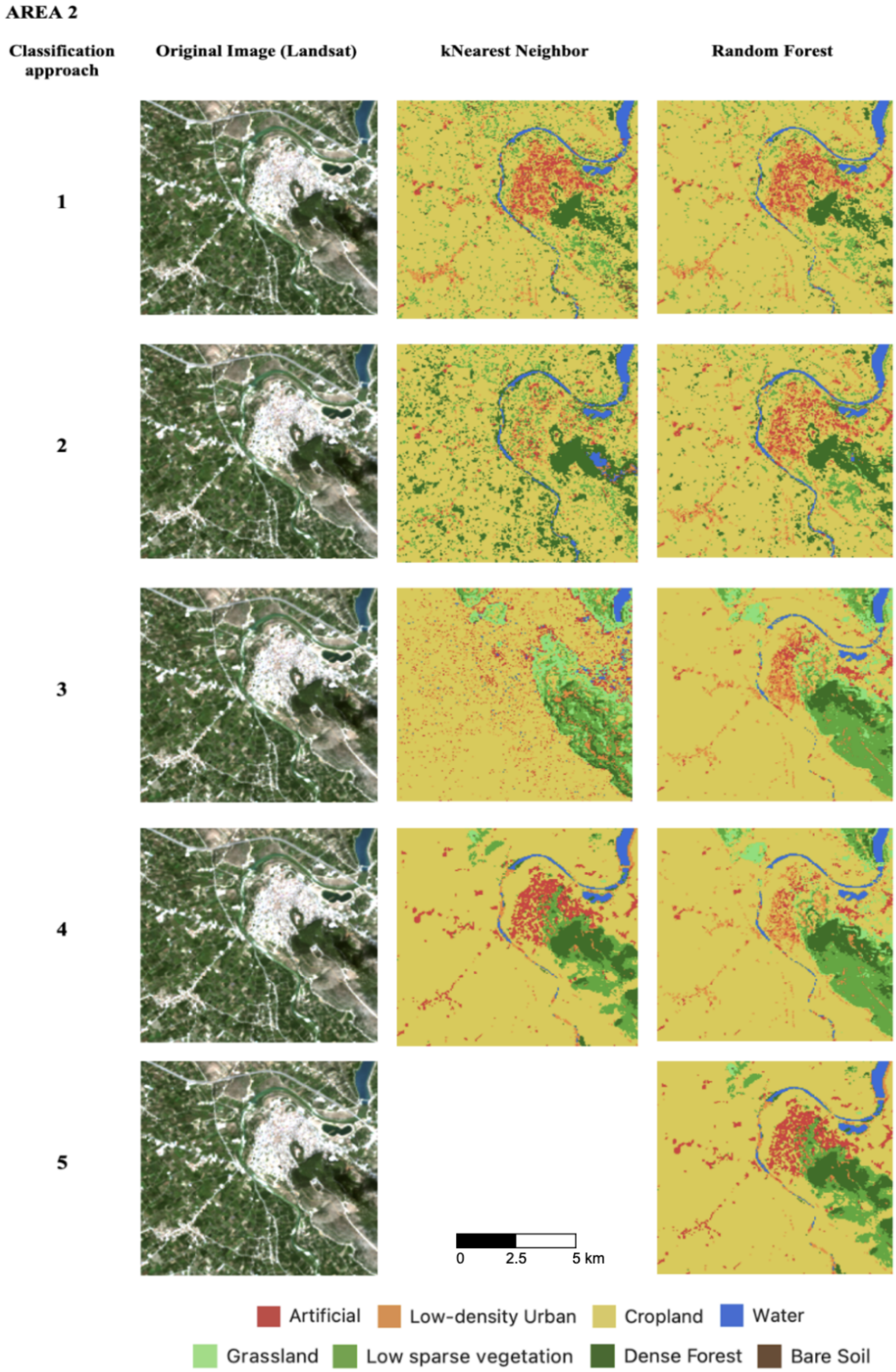


Figure 12B: Original Landsat-8 true color image (acquisition date: 12/05/2018, RGB = 432) and land cover classification results using kNearest Neighbor and Random Forests algorithms for sub-region 2.

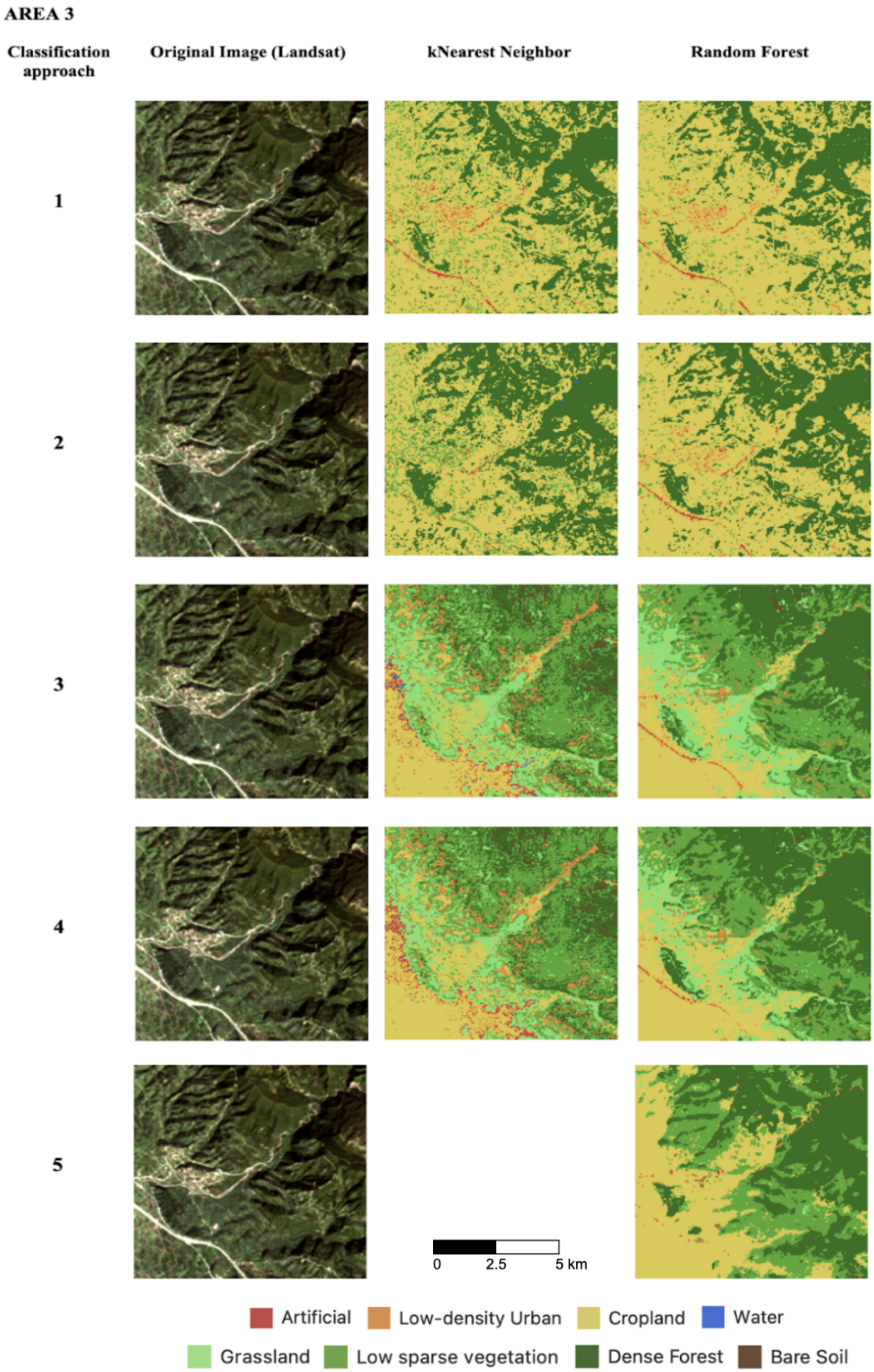


Figure 12C: Original Landsat-8 true color image (acquisition date: 12/05/2018, RGB =432) and land cover classification results using kNearest Neighbor and Random Forests algorithms for sub-region 3.

5. Discussion

In this study, the main goal is to investigate the benefit of a more complex classification method that takes as input a wider variety of satellite image datasets than multispectral data, for the land cover type estimation. The research questions that are 'Yes or No' questions are focused on the selected area of interest and are the following:

- Does the integration of the surface's thermal information with MS data lead to better classification results?
- Does the integration of the terrain's topography information with MS data lead to better classification results?
- Does the combination of all the above information lead to better classification results?
- Does the usage of the time series of all the above information lead to better classification results?

Five different classification approaches were designed, implemented and performed, having in each one different input datasets in order to generate land cover maps that include the following eight classes: *'Artificial'* surfaces, '*Bare Soil*', *'Cropland'*, '*Dense Forest*', *'Grassland'*, '*Low Density Urban*' areas, '*Low/Sparse Vegetation*', and *'Water'*. The approaches were:

1. Using only the multispectral bands of Landsat 8/9 satellite sensors,
2. Using both multispectral and thermal infrared bands of Landsat 8/9,
3. Using both multispectral bands and terrain information of the area, as derived from ASTER GDEM,
4. Using all the above - multispectral/thermal infrared bands and terrain information, and,
5. Using the time-series of all datasets.

The first approach was considered as reference for the extraction of the thesis results. Furthermore, the performance of two different machine learning algorithms was investigated on the classification of the above-mentioned classes: kNearest Neighbor and Random Forests.

According to the findings extracted from the visual photointerpretation and numerical comparison of the evaluation of the two algorithms, as presented in Chapter 3, very high accuracies were achieved for most classifications, reaching up to 0.98 for the OA and 1 for the Kappa, and for most classes. A further analysis was implemented in order to determine which algorithm generated the highest confusion between the selected classes. From the confusion matrices (Table 13A, Table 13B), the percentage of wrongly classified pixels to the total number of correctly classified pixels was calculated per class. From these, only these that indicate a percentage of more than 15% of the have been collected and summarized in Table 14.

Table 14: Confusions of each land cover class for the different classification approaches used in this study (Green: only in kNearest Neighbor algorithm, ***Black****: in both algorithms).*

Class	CA1	CA2	CA3	CA4	CA5
Artificial	-	-	-	Cropland (kNN: 71%)	-
Bare Soil	• Low/Sparse Vegetation (kNN: 30%, RF: 39%) • Cropland (kNN: 19%,	• Low/Sparse Vegetation (kNN: 37%, RF: 16%) • Cropland (kNN: 45%,	-	-	-

	RF: 24%)	RF: 27%)			
Cropland	-	-	-	-	-
Dense Forest	-	-	-	-	-
Grassland	Cropland (kNN: 183%, RF: 192%)	• Cropland (kNN: 220%, RF: 103%) • Low/Sparse Vegetation (kNN: 30%)	-	-	-
Low Density Urban	Cropland (kNN: 21%, RF: 24%)	• Cropland (kNN: 80%, RF: 23%) • Low/Sparse Vegetation (kNN: 36%)	-	Cropland (kNN: 34%)	-
Low/Sparse Vegetation	Cropland (kNN: 35%, RF: 38%)	Cropland (kNN: 53%, RF: 24%)	-	-	-
Water	-	-	-	-	-

Quantitative, both algorithms performed well when the inputs were both the multispectral bands and the terrain products, with a chance of a pixel to be classified correctly to be more than 85%. Studies that incorporated terrain data as input into the classification algorithms, however, showed lower accuracies (Liu et al, 2018, Hosseiny et al., 2020). Good performance is also observed for the classification using time-series data, a result that agrees with other relevant studies that achieve more than 88.9% overall accuracy when using time series for land cover classification (Simonetti, Simonetti, and Preatoni, 2014, and Schäfer et al, 2019).

Both algorithms did not perform well in CA1, where only multispectral bands were used as input. In most classes there was a chance of confusion of more than 19%. However, the kNearest Neighbor algorithm is observed to have better performance than Random Forests in this classification approach. The opposite is shown in CA2, where inputs were both multispectral and thermal infrared bands. For classes that presented high levels of confusion, Random Forests performed better than the kNearest Neighbor algorithm, with the latter also having worse performance for more classes (e.g. *'Grassland'* and '*Low Density Urban*' were largely confused with '*Low/Sparse Vegetation*' apart from *'Cropland'* with the kNearest Neighbor algorithm. Additionally, the Random Forests algorithm performed better in distinguishing artificial surfaces, as compared to kNearest Neighbor. Finally, for CA4, where all data were used as input, Random Forests showed good performance in terms of class confusion, whereas kNearest Neighbor had high levels of confusion in *'Artificial'* and '*Low Density Urban*'.

Taking all the above into consideration, it can be summarized that, in this study, between kNearest Neighbor and Random Forests machine learning classification algorithms:

- Both algorithms have shown good performance for the selected classes over the study area when having as input multispectral bands and terrain products (elevation, slope, TPI),
- Random Forests performs better when thermal information is included in the input datasets,

- The kNearest Neighbor algorithm had the lowest performance in this classification approach,
- The kNearest Neighbor generates better results than Random Forests when inputs are only multispectral data,
- The Random Forests algorithm performs better in distinguishing artificial surfaces compared to kNearest Neighbor, and
- The kNearest Neighbor algorithm appears to have significant speckle noise, whereas Random Forests classification results are more consistent.

From all the above results and findings from Chapter 3, and using as reference the classification results as generated using only multispectral data, the answers for the research questions set in this study are the following:

Question 1

Does the integration of the surface's thermal information with MS data lead to better classification results?

After computing the performance change between the classification approach 2 compared to the reference approach, the percentile difference of the accuracy metrics was calculated. The % difference of the Overall Accuracy (OA) and the Kappa coefficient equals to -2.8% and -4.9% for the kNearest Neighbors classification results, and 1.7% and 3.3% for the Random Forests classification respectively. This means that, overall, when surface thermal infrared information is added to the multispectral bands while performing a supervised classification using kNearest Neighbor classifier, the result of this study shows no better performance than using only the multispectral bands. On the contrary, when using a Random Forests classifier, the resulting accuracies perform slightly better. Liya et al. (2015) conducted a similar study using Landsat 4/5 images. When the thermal bands of Landsat were added to the multispectral bands for a land cover classification, there was an increase of 3-6% in the OA, slightly bigger than the improvement calculated in this study. However, kNearest Neighbor classifier performed better than the Random Forests (Liya et al, 2015)- something that in this study was not achieved.

Table 15: Percentage difference of the User and Producer Accuracy metrics for each class as generated from kNearest Neighbors and Random Forests classifiers for the performance assessment between the Classification Approach 2 (CA2) and the Classification Approach 1 (reference) used in this study.

Class	kNearest Neighbors (n=5)		Random Forests (n=100)	
	% User Accuracy difference	% Producer Accuracy difference	% User Accuracy difference	% Producer Accuracy difference
Artificial	-28	-9	0	-4
Bare Soil	-28	-21	12	8
Cropland	-2	-4	2	4
Dense Forest	-4	-4	2	2
Grassland	-19	-15	32	15

Low Density Urban	-84	-29	1	1
Low/Sparse Vegetation	-16	-15	12	9
Water	0	0	0	0
TOTAL	*OA = -2.8%* *Kappa = -4.9%*		*OA = 1.7%* *Kappa = 3.3%*	

The percentile differences for the accuracy metrics of each class are presented in Table 15. The classes whose accuracy metrics (User Accuracy, Producer Accuracy) presented the biggest decreases are Low Density Urban (with a decrease of -84% and -29% respectively), Bare Soil (with -28% and -21% respectively), and Artificial (with -28% and -9% respectively).

On the contrary, when using a Random Forests classifier, the resulting accuracies performed slightly better, depending on the class. The classes whose accuracy metrics (User Accuracy, Producer Accuracy) presented the biggest increases are: Grassland (with an improvement of +32% and +15% respectively), Low/Sparse Vegetation (improved by +12% and +9% respectively), and Bare Soil (with +12% and +8% respectively).

Of the classes selected in this study, only the ‘Water’ didn’t show any change in its performance. This may be due to the fact that water bodies were already clearly discriminated from the classification using as input the multispectral information, due to absence of spectral mixtures (Sinha et al, 2015).

Question 2

Does the integration of the terrain’s topography information with MS data lead to better classification results?

After computing the performance change between the classification approach 3 compared to the reference approach, the percentile difference of the accuracy metrics was calculated. The % difference of the OA and the Kappa coefficient equals to 6.5% and 13.0% for the kNearest Neighbors classification results, and 5.8% and 11.8% for the Random Forests classification respectively. This means that, overall,when adding the terrain’s topography information to multispectral bands while performing a supervised classification, the output has better results when using either kNearest Neighbor classifier or Random Forests. This comes to support the outcome of other studies that demonstrate this improvement having as input terrain information, as well, even though using other algorithms (Liu et al, 2018, Sang et al, 2021, Jwan et al, 2022).

Table 16: Percentage difference of the User and Producer Accuracy metrics for each class as generated from kNearest Neighbors and Random Forests classifiers for the performance assessment between the Classification Approach 3 and the Classification Approach 1 (reference) used in this study.

Class	**kNearest Neighbors (n=5)**	**Random Forests (n=100)**

	% User Accuracy difference	% Producer Accuracy difference	% User Accuracy difference	% Producer Accuracy difference
Artificial	11	13	10	11
Bare Soil	27	30	39	18
Cropland	7	13	5	13
Dense Forest	6	4	5	4
Grassland	68	34	67	24
Low Density Urban	24	18	22	14
Low/Sparse Vegetation	31	27	30	22
Water	0	0	0	0
TOTAL	*OA = 6.5%* *Kappa = 13.0%*		*OA = 5.8%* *Kappa = 11.8%*	

The classes whose accuracy metrics (User Accuracy, Producer Accuracy) present the biggest increases with kNearest Neighbor areGrassland (with +68% and +34% respectively), Low/Sparse Vegetation (improved by +31% and +27% respectively), and Bare Soil (with +27% and +30% respectively), while for those generated with Random Forests are: Grassland (+67% and +24% respectively), Bare Soil (+39% and +18% respectively), and Low/Sparse Vegetation (with an improvement of +30% and +22% respectively).

Of the classes selected in this study, only the metrics of 'Water' didn't show any change.

<u>Question 3</u>

Does the combination of all the above information lead to better classification results?

The percentile difference of the accuracy metrics between classification approach 4 and 1 (reference) was calculated. The results show that the OA and the Kappa coefficient equals to 5.3% and 10.7% for the kNearest Neighbors classification results, and 5.8% and 11.8% for the Random Forests classification respectively. Compared to the accuracy improvement in the previous two research questions that referred to the integration of thermal and elevation products separately into a land cover classification from multispectral data, also in this one, when adding both surface thermal infrared bands and the terrain's topography information to multispectral bands while performing a supervised classification, the output has again better results than using only multispectral data Rehman et al. (2021) performed a land cover classification using Random Forests classifier on Landsat-8 imagery and products, and investigated the impact of adding elevation and land surface temperature data in the algorithm. Their results showed an even higher improvement than the results of this study: an increase of 20% in the OA and 33% in the Kappa coefficient. This suggests that ancillary variables carry significant significance in the classification process and should be considered in conjunction with spectral bands.

Table 17: Percentage difference of the User and Producer Accuracy metrics for each class as generated from kNearest Neighbors and Random Forests classifiers for the performance assessment between the Classification Approach 4 and the Classification Approach 1 (reference) used in this study.

Class	kNearest Neighbors (n=5)		Random Forests (n=100)	
	% User Accuracy difference	% Producer Accuracy difference	% User Accuracy difference	% Producer Accuracy difference
Artificial	-54	5	9	10
Bare Soil	32	29	39	18
Cropland	7	10	5	13
Dense Forest	4	3	5	4
Grassland	65	33	67	24
Low Density Urban	-6	7	21	14
Low/Sparse Vegetation	29	25	30	22
Water	0	0	0	0
TOTAL	*OA = 5.3%* *Kappa = 10.7%*		*OA = 5.8%* *Kappa = 11.8%*	

The classes whose accuracy metrics (User Accuracy, Producer Accuracy) present the biggest increases with kNearest Neighbor are Grassland (with an improvement of +65% and +33% respectively), Bare Soil (with +32% and +29% respectively), and Low/Sparse Vegetation (improved by +29% and +25% respectively), while for those generated with Random Forests are Grassland (+67% and +24% respectively), Bare Soil (+39% and +18% respectively), and Low/Sparse Vegetation (with +30% and +22% respectively).

Of the classes selected in this study, only the metrics of 'Water' didn't show any change. Furthermore, it should be noted that the result generated using kNearest Neighbor algorithm presented a lowered User Accuracy for the 'Artificial' class, with a -54% decrease from using only multispectral data as input. Also, this result generated with Random Forests performed as well as the equivalent result from the previous research question.

Question 4

Does the usage of the of all the above information lead to better classification results?

Time series of the multispectral bands, the thermal infrared bands and the terrain's topography information in a supervised classification, was performed only with the Random Forests classifier, due to performance restrictions of the kNearest Neighbor classifier. The percentile difference of the accuracy metrics between classification approach 5 and 1 (reference) was calculated. The results indicate a noticeable improvement when incorporating time series data into the land cover classification of this study. Specifically, the OA increased by 5.8%, while the Kappa coefficient saw a significant rise of 12.3%. These findings underscore the positive impact of time series data on the classification performance.

In a similar context, Amini et al. (2022) conducted a Random Forests-based land cover classification. They, too, integrated Landsat time series data along with thermal bands and elevation information as input features. Notably, their study reported even more substantial improvements, with an 11.9% increase in OA and a substantial 17.7% boost in the Kappa coefficient. These results demonstrate the considerable advantage of incorporating time series data, thermal bands, and elevation information in land cover classification, reaffirming its potential for enhancing accuracy in such applications.

Table 18: Percentage difference of the User and Producer Accuracy metrics for each class as generated from Random Forests classifiers for the performance assessment between the Classification Approach 5 and the Classification Approach 1 (reference) used in this study.

Class	**Random Forests (n=100)**	
	% User Accuracy difference	**% Producer Accuracy difference**
Artificial	7	9
Bare Soil	40	20
Cropland	5	14
Dense Forest	6	5
Grassland	68	25
Low Density Urban	23	12
Low/Sparse Vegetation	31	23
Water	0	0
TOTAL	*OA = 5.8%* *Kappa = 12.3%*	

The classes whose accuracy metrics (User Accuracy, Producer Accuracy) showed the biggest increases were Grassland (with an improvement of +68% and +25% respectively), Bare Soil (with +40% and +20% respectively), Low/Sparse Vegetation (improved by +29% and +25% respectively), and Low Density Urban (improved by +23% and +22% respectively).

According to Amini et al. (2022), classes may be affected from height patterns, which assist in the increase of the final classification accuracy. In their study, the classes that performed better were the ‘Bare Land’, and ‘Shrub’, which thematically correspond to the abovementioned (i.e. Bare Soil and Low/Sparse Vegetation). Of the classes selected in this study, only the metrics of ‘Water’ didn’t show any change also in this comparison. It should be noted that the generated Random Forests result performed slightly better than the equivalent results from the previous research questions.

6. Conclusions

The aim of the project was the classification of land cover types in the Ionian Sea region in Greece. The study examined different methods of processing and combining remote sensing data from different sensors, using the kNearest Neighbors and Random Forests supervised machine learning techniques, a total of eighty-eight (88) Landsat 8 and Landsat 9 multispectral and thermal imagery scenes within a ten year span (2013-2022), and topography information from the ASTER GDEM. Data variability over time through the generation of time series dataset was also considered. Eight different land cover classes over the study area were depicted, using as ground truth the 2018 CORINE Land Cover product in combination with photo interpretation, with more than 14,000 training pixel samples per class retrieved from the entire dataset.

A holistic approach was followed by combining the abovementioned different datasets in different classification methodologies. Questions addressed included the effect of thermal properties, elevation and topography on classification, as well as the use of time series for improved results compared to using only multispectral data. The aim was not only to investigate the effectiveness of this multidimensional approach, but also to determine whether it actually leaded to a noticeable improvement in the quality of land cover classification results for the study area selected. The findings showed that when multispectral data were combined with either terrain information, thermal infrared bands, or both, the classification results improved satisfactorily with both kNearest Neighbor and Random Forests classifiers. This improvement reached up to 6.5% in the OA and 11.8% in the Kappa coefficient. Best performance in the classification output was calculated when time-series information of all the above were incorporated as input in the Random Forests classifier. The level of the enhancement reached up to 68% on specific classes, mostly relevant to vegetation.

The results presented above validate findings in existing literature. Over the years, numerous research studies have tackled the challenge of land cover classification from high-resolution satellite data, utilizing input datasets that correspond to those used in this study. Thus, conducting this thorough analysis further contributed to the ongoing debate in the field and shed light on the potential benefits of integrating different data sources for more accurate land cover classification.

Future work on this study could include the investigation on some of the following topics:

- Seasonality of specific classes, e.g. croplands, deciduous forests, grasslands
- Elevation of certain classes, specifically related to vegetation (forests, sparse vegetation, etc) that is dependent also to the study area's climate flora
- Irregular changes in land cover, for example expansion of urban areas, new construction sites, reforestation/deforestation, wildfires, need to be taken into consideration before performing a land cover classification
- Detection of clouds and shadowed areas over the study area, and elimination from the classification process
- Experimentation with different Random Forests and kNearest Neighbors parameters and sample numbers

References

Abdi, M. A. 2019. Land Cover and Land Use Classification Performance of Machine Learning Algorithms in a Boreal Landscape Using Sentinel-2 Data. *GIScience & Remote Sensing*. DOI: 10.1080/15481603.2019.1650447.

Amini, S., Saber, M., Rabiei-Dastjerdi, H., & Homayouni, S. 2022. Urban land use and land cover change analysis using random forest classification of Landsat time series. *Remote Sensing*, 14(11), 2654.

ArcGIS Pro, 2023, *How slope works*, accessed 20 May 2023, <https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/how-slope-works.htm>.

Breiman, L. 1984. Classification and Regression Trees. *Boca Raton: Chapman & Hall/CRC*. ISBN 978-0-412-04841-8.

Campbell, J.B., Wynne, R.H., and Thomas, V.A. 2011. Introduction to Remote Sensing, Fifth Edition, ISBN: 9781609181765

Charou, E., Felekis, G., Bournou Stavroulopoulou, D., Koutsoukou, M., Panagiotopoulou, A., Voutos, Y., Bratsolis, E., Mylonas, P., Likforman-Sulem, L. 2019. Deep Learning for Agricultural Land Detection in Insular Areas. *IEEE* 978-1-7281-4959-2/19.

Chen, Y., Tang, S., Bouguila, N., Wang, C., Du, J. and Li, H., 2018. A fast clustering algorithm based on pruning unnecessary distance computations in dbscan for high-dimensional data, *Pattern Recognition*, vol. 83, p. 375-387.

Copernicus Land Monitoring Service, 2023, *CORINE Land Cover*, accessed 20 May 2023, <https://land.copernicus.eu/pan-european/corine-land-cover>.

European Space Agency - ESA, 2023, *Sentinel-2*, accessed 20 May 2023, <https://sentinel.esa.int/web/sentinel/missions/sentinel-2>.

European Space Agency - ESA-Eduspace, 2009, *Spectral signatures*, accessed 20 May 2023, <https://www.esa.int/SPECIALS/Eduspace_EN/SEMPNQ3Z2OF_0.html>.

Forkuor, G., Dimobe, K., Serme, I. & Tondoh, J.E. 2018. Landsat-8 vs. Sentinel-2: examining the added value of sentinel-2's red-edge bands to land-use and land-cover mapping in Burkina Faso, *GIScience & Remote Sensing*, 55:3, 331-354, DOI: 10.1080/15481603.2017.1370169.

Gounaridis, D., Apostolou, A., & Koukoulas, S. 2016. Land Cover of Greece, 2010: a Semi-Automated Classification Using Random Forests. *Journal of Maps* 12:5, 1055-1062. DOI: 10.1080/17445647.2015.1123656.

Han, H., Zeng, Q. and Jiao, J., 2021. Quality assessment of TanDEM-X DEMs, SRTM and ASTER GDEM on selected Chinese sites. *Remote Sensing*, 13(7), p.1304.

Ho, T.K. 2002. "A Data Complexity Analysis of Comparative Advantages of Decision Forest Constructors", *Pattern Analysis and Applications*, 5 (2): 102–112.

Hosseiny, B., Abdi, A.M. and Jamali, S., 2022. Urban land use and land cover classification with interpretable machine learning–A case study using Sentinel-2 and auxiliary data. *Remote Sensing Applications: Society and Environment*, *28*, p.100843.

Hostert, P., Griffiths, P., van der Linden, S. and Pflugmacher, D. 2015. Time Series Analyses in a New Era of Optical Satellite Data. *Remote Sensing Time Series, Springer*, 25–41.

Humbolt State University, 2019, *Accuracy Metrics*, accessed 20 May 2023, <http://gsp.humboldt.edu/olm/Courses/GSP_216/lessons/accuracy/metrics.html#:~:text=User's%20Accuracy,-The%20User's%20Accuracy&text=This%20is%20referred%20to%20as,it%20by%20the%20row%20total>.

Hurskainen, P., Adhikari, H., Siljander, M., Pellikka, P.K.E. and Hemp, A., 2019. Auxiliary datasets improve accuracy of object-based land use/land cover classification in heterogeneous savanna landscapes. *Remote sensing of environment*, 233, p.111354.

IBM, 2023, *K-Nearest Neighbors Algorithm*, accessed 20 May 2023, <https://www.ibm.com/topics/knn>.

Jwan Al-Doski, Faez M. Hassan, Hussein Abdelwahab Mossa, and Aus A. Najim. 2022. Incorporation of Digital Elevation Model, Normalized Difference Vegetation Index, and Landsat-8 Data for Land Use Land Cover Mapping. Photogrammetric Engineering & Remote Sensing, Vol. 88, No. 8, pp. 507–515. 0099-1112/22/507–515

Kennedy, R.E., Ziquiang, Y., Cohen, W., 2010. Detecting trends in forest disturbance and recovery using yearly Landsat time series: 1. LandTrendr – temporal segmentation algorithms. Remote Sening of the Environment, 114, pp. 2897-2910.

Knitter, D., Brozio, J.P., Hamer, W., Duttmann, R., Müller, J. and Nakoinz, O., 2019. Transformations and site locations from a landscape archaeological perspective: The case of Neolithic Wagrien, Schleswig-Holstein, Germany. *Land*, 8(4), p.68.

Liu, Y.; Gong, W.; Hu, X. and Gong, J. 2018. Forest Type Identification with Random Forest Using Sentinel-1A, Sentinel-2A, Multi-Temporal Landsat-8 and DEM Data. *Remote Sensing,* 10, 946; doi:10.3390/rs10060946.

Liya, S. and Schulz, K. 2015. The Improvement of Land Cover Classification by Thermal Remote Sensing. *Remote Sensing* 7, 8368-8390. 10.3390/rs70708368.

Manandhar, R., Odeh, I.O. and Ancev, T., 2009. Improving the accuracy of land use and land cover classification of Landsat data using post-classification enhancement. Remote Sensing, 1(3), pp.330-344.

NASA/METI/AIST/Japan Spacesystems, and U.S./Japan ASTER Science Team. 2018. ASTER Global Digital Elevation Model V003, distributed by NASA EOSDIS Land Processes DAAC, https://doi.org/10.5067/ASTER/ASTGTM.003

Olthof, I., Fraser, R.H. 2014. Detecting landscape changes in high latitude environments using Landsat trend analysis: 2. Classification. Remote Sens., 6 (11) (2014), pp. 11558-11578.

Orfeo ToolBox, 2023, Superimpose, accessed 16 October 2023, <https://www.orfeo-toolbox.org/CookBook/Applications/app_Superimpose.html>

Phiri, D., & Morgenroth, J. 2017. Developments in Landsat land cover classification methods: A review. *Remote Sensing*, 9(9), 967.

Rana, V.K., Suryanarayana, T.M.V. 2019. Visual and statistical comparison of ASTER, SRTM, and Cartosat digital elevation models for watershed. *Journal of geovisualization and spatial analysis*, 3, 12. https://doi.org/10.1007/s41651-019-0036-z.

Romero, A., Gatta, C. and Camps-Valls, G. 2016. Unsupervised Deep Feature Extraction for Remote Sensing Image Classification, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 3, pp. 1349-1362, doi: 10.1109/TGRS.2015.2478379.

Samuel, A.L., 1959. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3), pp.210-229.

Sang, X., Guo, Q., Wu, X. et al. 2021. The Effect of DEM on the Land Use/Cover Classification Accuracy of Landsat OLI Images. Journal of the Indian Society of Remote Sensing, 49, 1507–1518. https://doi.org/10.1007/s12524-021-01318-5

Satgé, F., Bonnet, M.P., Timouk, F., Calmant, S., Pillco, R., Molina, J., Lavado-Casimiro, W., Arsen, A., Crétaux, J.F. and Garnier, J., 2015. Accuracy assessment of SRTM v4 and ASTER GDEM v2 over the Altiplano watershed using ICESat/GLAS data. *International Journal of Remote Sensing*, 36(2), pp.465-488.

Schäfer, P., Pflugmacher, D., Hostert, P., Leser, U. 2019. Classifying Land Cover from Satellite Images Using Time Series Analytics. *Conference Paper, 2nd International Workshop on Data Analytics Solutions for Real-Life Applications,* Vienna, Austria.

Simón Sánchez, A.M., González-Piqueras, J., de la Ossa, L. and Calera, A. 2022. Convolutional Neural Networks for Agricultural Land Use Classification from Sentinel-2 Image Time Series. *Remote Sensing*, 14(21), p.5373.

Simonetti, E., Simonetti, D., Preatoni, D. 2014. Phenology-Based Land Cover Classification Using Landsat 8 Time Series. *Technical Report by the Joint Research Centre of the European Commission,* ISBN 978-92-79-40844-1. doi: 10.2788/15561.

Sinha, S., Sharma, L. K., Nathawat. M. S. 2015. Improved Land-use/Land-cover classification of semi-arid deciduous forest landscape using thermal remote sensing. The Egyptian Journal of Remote Sensing and Space Sciences, 18, 217–233.

Sohl, T.L., Sleeter, B.M., Zhu, Z., Sayler, K. L., Bennett, S., Bouchard, M., Reker, R., Hawbaker, T., Wein, A., Liu, Sh., Kanengieter, R., Acevedo, W. 2012. A land-use and land-cover modeling strategy to support a national assessment of carbon stocks and fluxes, *Applied Geography*, Volume 34, 2012, Pages 111-124, ISSN 0143-6228, https://doi.org/10.1016/j.apgeog.2011.10.019.

Svoboda, J., Štych, P., Laštovička, J., Paluba, D., Kobliuk, N. 2022. Random Forest Classification of Land Use, Land-Use Change and Forestry (LULUCF) Using Sentinel-2 Data - A Case Study of Czechia. Remote Sens., 14, 1189. https://doi.org/10.3390/rs14051189

Talukdar, S., Singha, P., Mahato, S., Shahfahad Pal, S., Liou, Y.-A., Rahman, A. 2020. Land-Use Land-Cover Classification by Machine Learning Classifiers for Satellite Observations - A Review. Remote Sens., 12, 1135. https://doi.org/10.3390/rs12071135

Thakur, R., & Panse, P. 2022. Classification Performance of Land Use from Multispectral Remote Sensing Images using Decision Tree, K-Nearest Neighbor, Random Forest and Support Vector Machine Using EuroSAT Data. *International Journal of Intelligent Systems and Applications in Engineering*, 10(1s), 67–77.

Tulbure, M.G., Broich, M., 2013. Spatiotemporal dynamic of surface water bodies using Landsat time-series data from 1999 to 2011. ISPRS J. Photogramm. Remote Sens., 79, pp. 44-52

Ur Rehman, A., Ullah, S., Shafique, M. et al., 2021. Combining Landsat-8 spectral bands with ancillary variables for land cover classification in mountainous terrains of northern Pakistan. Journal of Mountain Science, 18, 2388–2401. https://doi.org/10.1007/s11629-020-6548-7

USGS, 2023, *ASTGTM v003*, accessed 20 May 2023, <https://lpdaac.usgs.gov/products/astgtmv003/>.

USGS, 2023, *Landsat Satellite Missions*, accessed 20 May 2023, <https://www.usgs.gov/core-science-systems/nli/landsat/landsat-satellite-missions>.

Weiss, A.D. 2001. Topographic position and landforms analysis. *Poster Presentation, ESRI Users Conference*, *San Diego, CA*.

Wilson, J.P. and Gallant, J.C. eds., 2000. Terrain analysis: principles and applications. *John Wiley & Sons*.

Yao, J., Chao-lu, Y. and Ping, F., 2020. Evaluation of the Accuracy of SRTM3 and ASTER GDEM in the Tibetan Plateau Mountain Ranges. In *E3S Web of Conferences* (Vol. 206, p. 01027). EDP Sciences.

Yuan, F., Sawaya, K. E., Loeffelholz, B. C., & Bauer, M. E. 2005. Land cover classification and change analysis of the Twin Cities (Minnesota) Metropolitan Area by multitemporal Landsat remote sensing. *Remote sensing of Environment*, 98(2-3), 317-328.

Yuh, Y., G., Tracz, W., Matthews, H. D., Turner, S. E. 2023. Application of machine learning approaches for land cover monitoring in northern Cameroon. Ecological Informatics, Volume 74, 101955, ISSN 1574-9541, https://doi.org/10.1016/j.ecoinf.2022.101955.

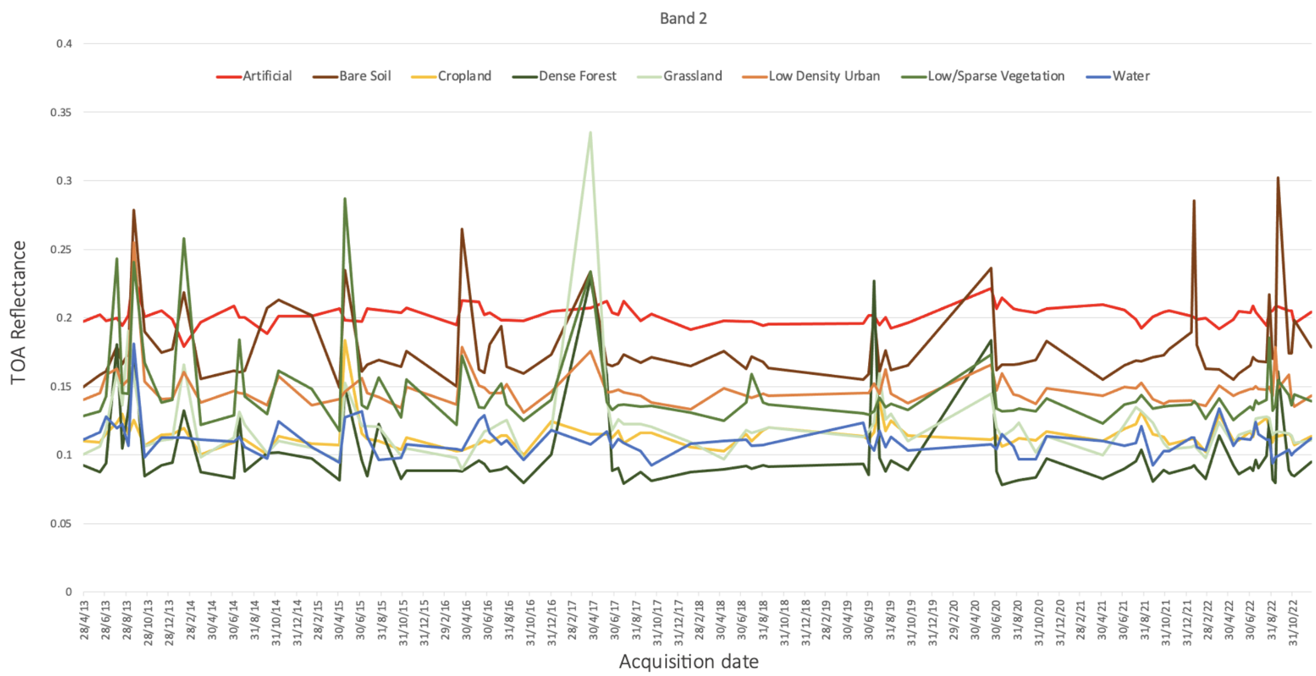
Zhu, Z., Woodcock, C.E. 2014. Automate cloud, cloud shadow, and snow detection in multitemporal Landsat data: an algorithm designed specifically for monitoring land cover change. Remote Sens. Environ., 152, pp. 217-234.

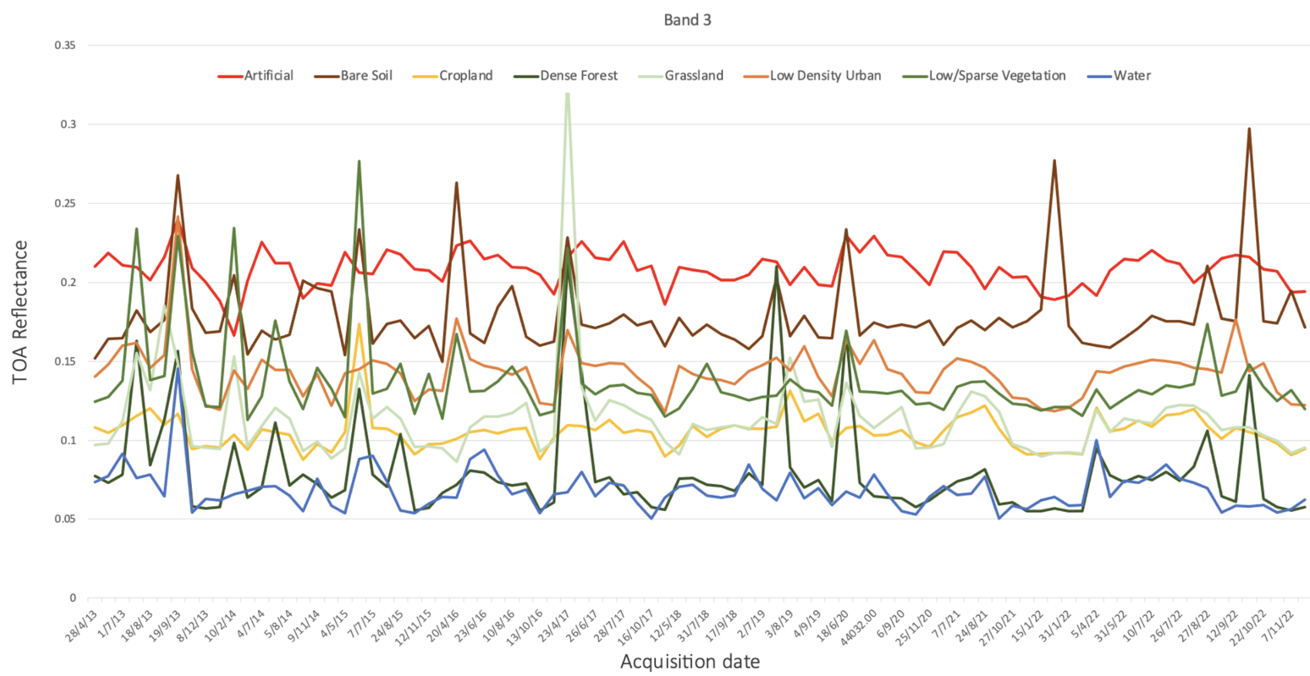
Annex A - Corine Land Cover nomenclature

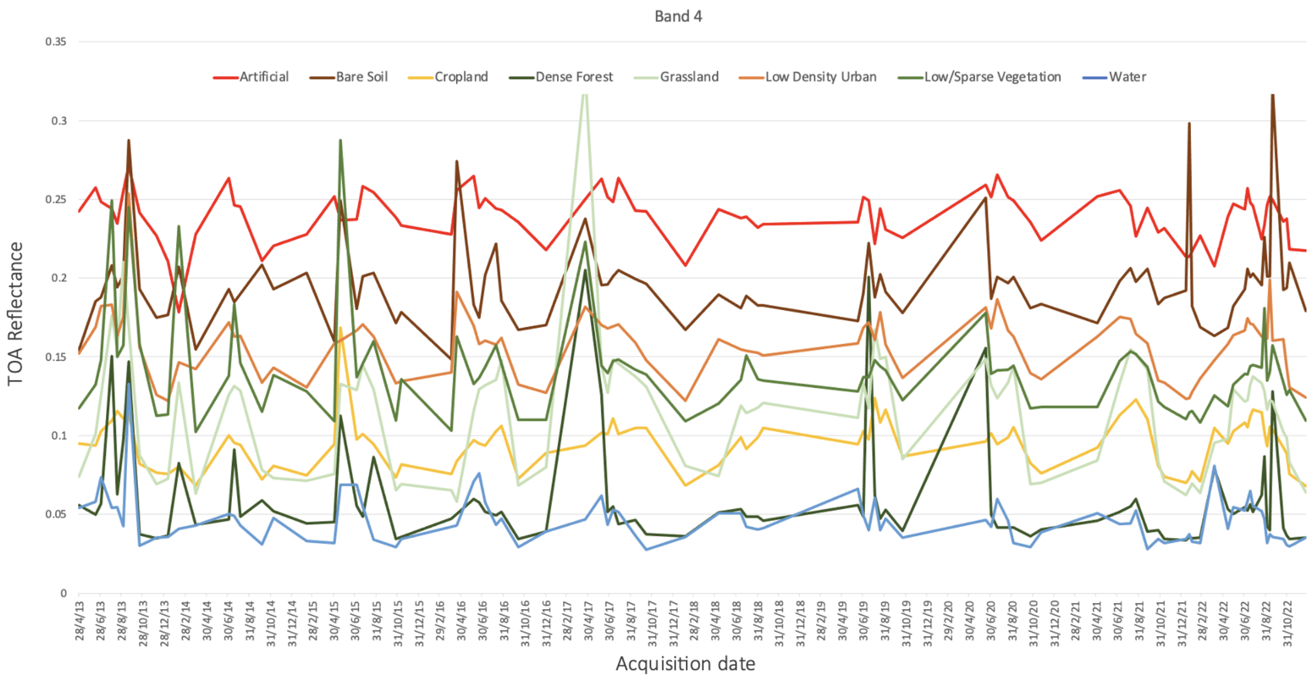
Table 1 – CORINE Land Cover (CLC) nomenclature (Source: http://www.igeo.pt/gdr/pdf/CLC2006_nomenclature_addendum.pdf).

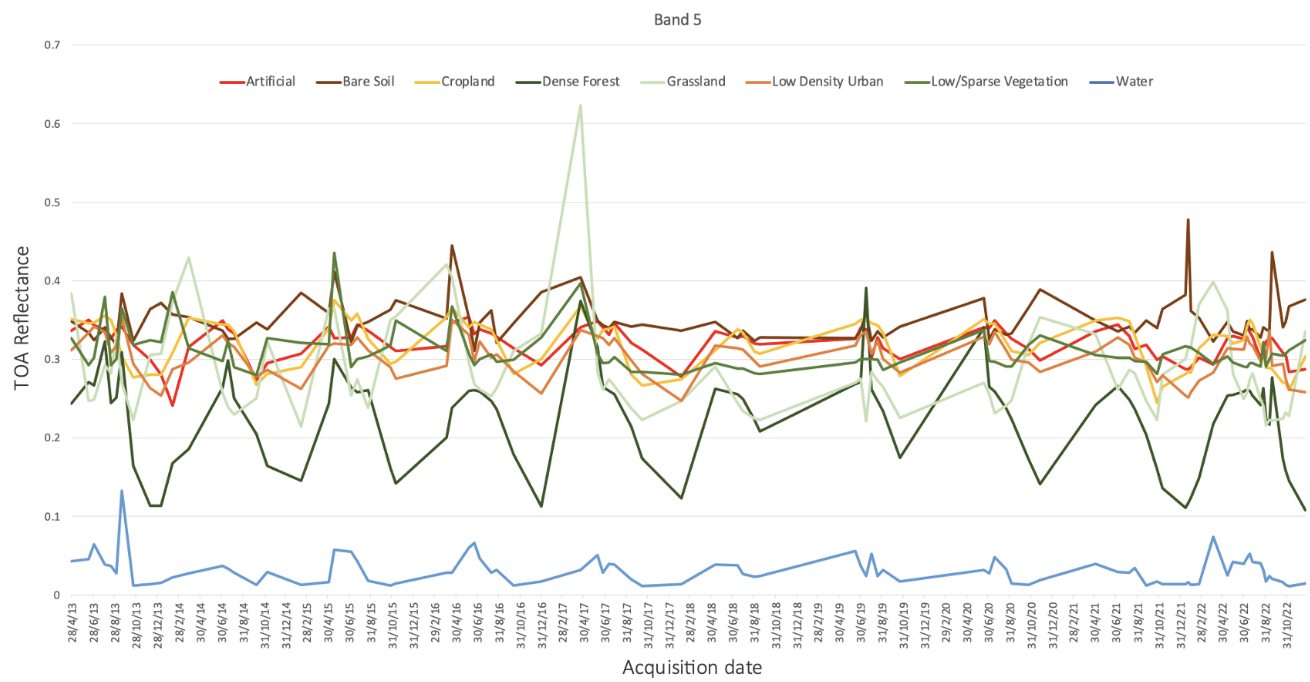
Level 1	Level 2	Level 3
1 Artificial surfaces	11 Urban fabric	111 Continuous urban fabric
		112 Discontinuous urban fabric
	12 Industrial, commercial and transport units	121 Industrial or commercial units
		122 Road and rail networks and associated land
		123 Port areas
		124 Airports
	13 Mine, dump and construction sites	131 Mineral extraction sites
		132 Dump sites
		133 Construction sites
	14 Artificial, non-agricultural vegetated areas	141 Green urban areas
		142 Sport and leisure facilities
2 Agricultural areas	21 Arable land	211 Non-irrigated arable land
		212 Permanently irrigated land
		213 Rice fields
	22 Permanent crops	221 Vineyards
		222 Fruit trees and berry plantations
		223 Olive groves
	23 Pastures	231 Pastures
	24 Heterogeneous agricultural areas	241 Annual crops associated with permanent crops
		242 Complex cultivation patterns
		243 Land principally occupied by agriculture, with significant areas of natural vegetation
		244 Agro-forestry areas
3 Forest and semi natural areas	31 Forests	311 Broad-leaved forest
		312 Coniferous forest
		313 Mixed forest
	32 Scrub and/or herbaceous vegetation associations	321 Natural grasslands
		322 Moors and heathland
		323 Sclerophyllous vegetation
		324 Transitional woodland-shrub
	33 Open spaces with little or no vegetation	331 Beaches, dunes, sands
		332 Bare rocks
		333 Sparsely vegetated areas
		334 Burnt areas
		335 Glaciers and perpetual snow
4 Wetlands	41 Inland wetlands	411 Inland marshes
		412 Peat bogs
	42 Maritime wetlands	421 Salt marshes
		422 Salines
		423 Intertidal flats
5 Water bodies	51 Inland waters	511 Water courses
		512 Water bodies
	52 Marine waters	521 Coastal lagoons
		522 Estuaries
		523 Sea and ocean

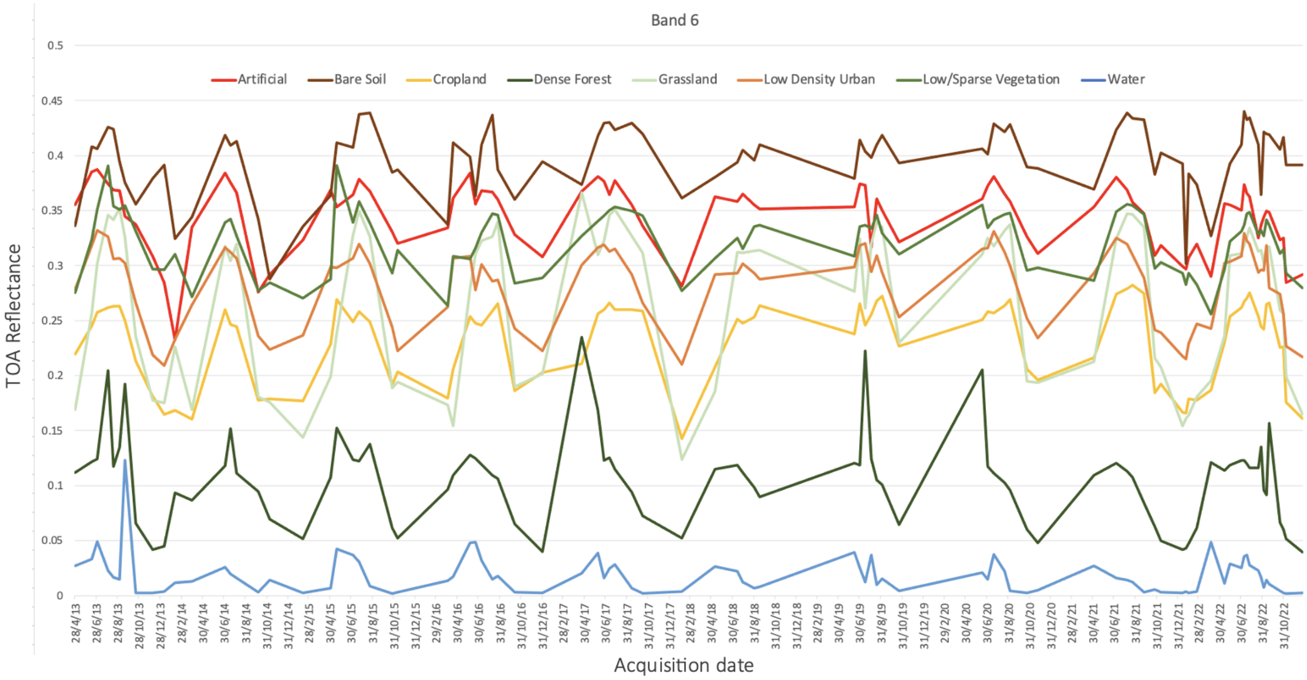
Annex B - Time series of the average pixel values per class and per band of Landsat scenes



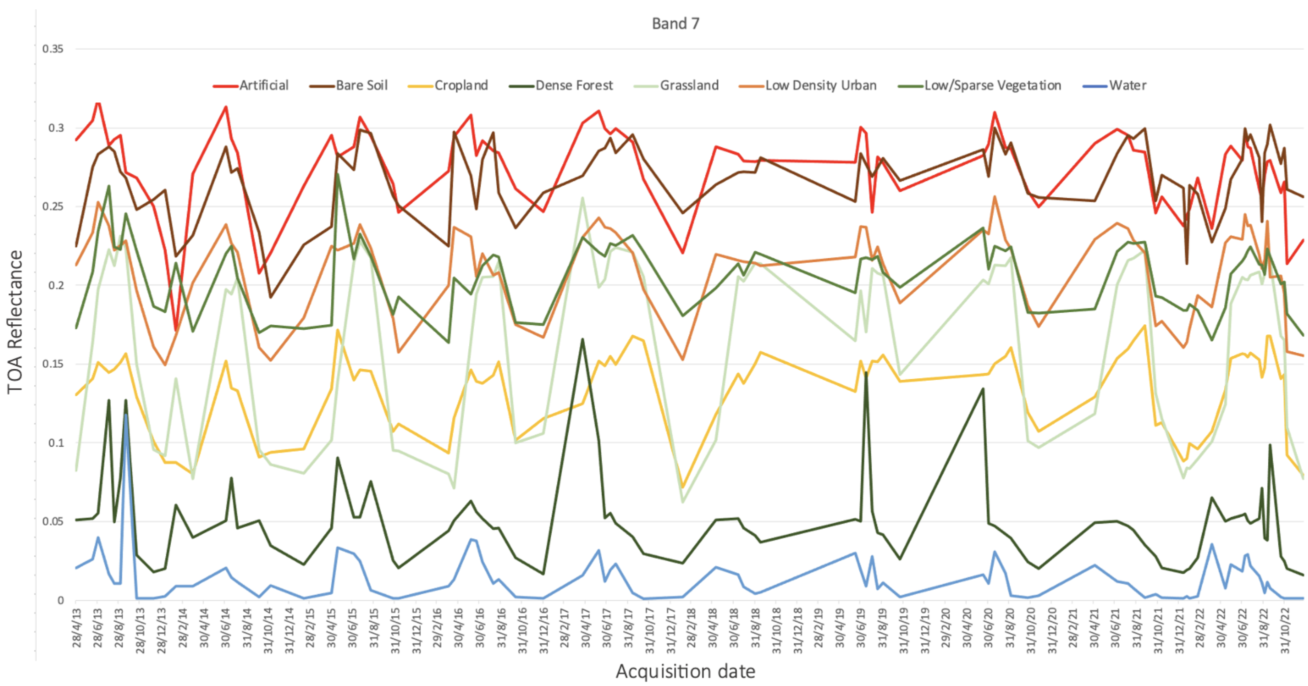



Band 4
Artificial
Bare Soil
Cropland
Dense Forest
Grassland
Low Density Urban
Low/Sparse Vegetation
Water
TOA Reflectance
Acquisition date

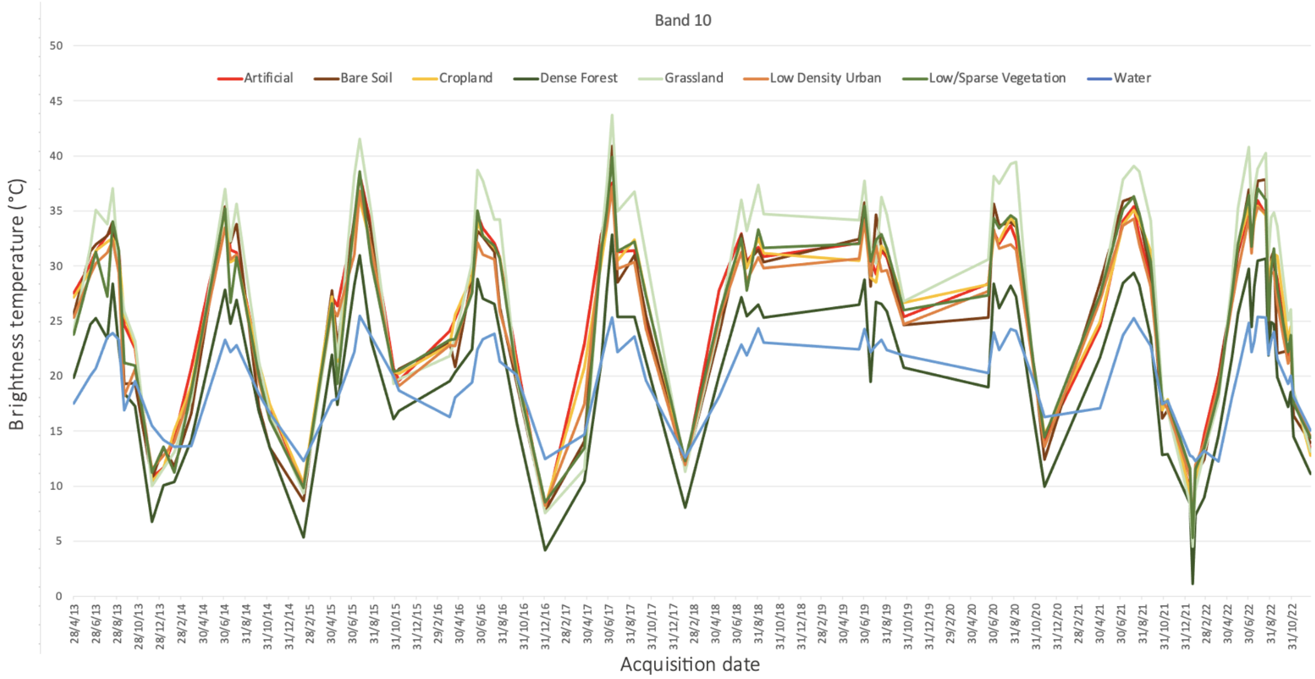

Band 5
Artificial
Bare Soil
Cropland
Dense Forest
Grassland
Low Density Urban
Low/Sparse Vegetation
Water
TOA Reflectance
Acquisition date



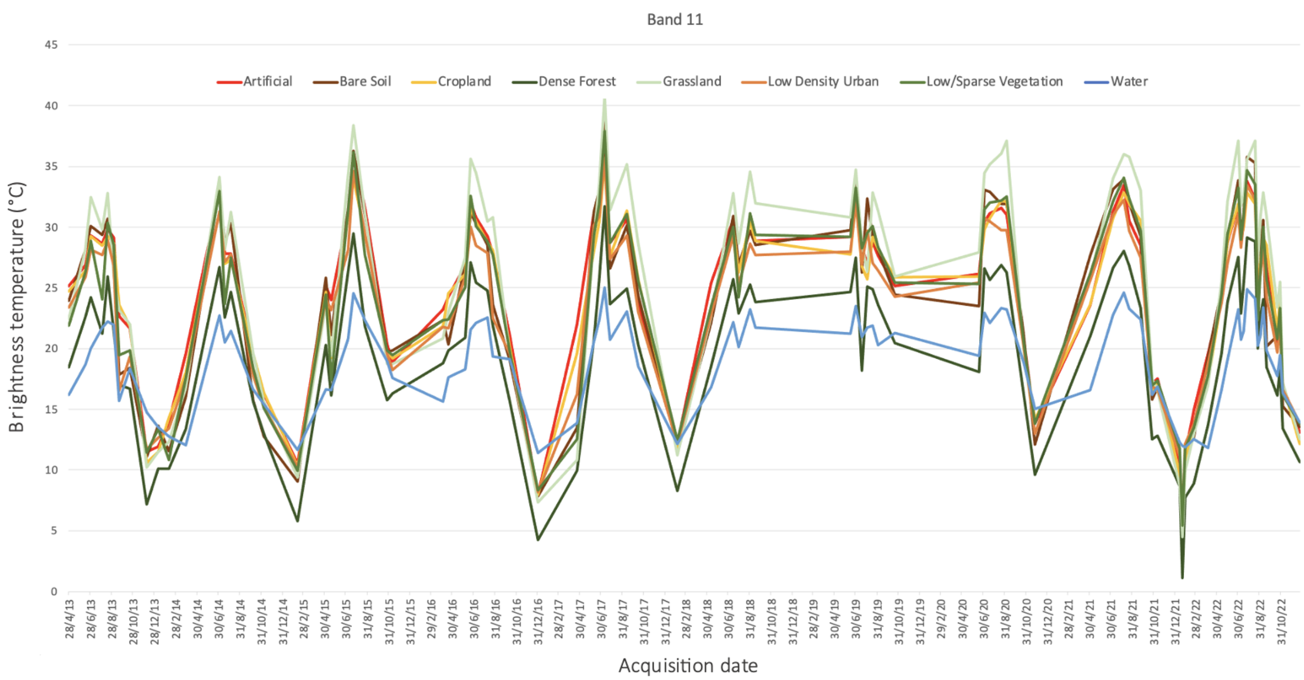
Band 6
Artificial
Bare Soil
Cropland
Dense Forest
Grassland
Low Density Urban
Low/Sparse Vegetation
Water
TOA Reflectance
Acquisition date



Band 7
Artificial
Bare Soil
Cropland
Dense Forest
Grassland
Low Density Urban
Low/Sparse Vegetation
Water
TOA Reflectance
Acquisition date



Band 10
Artificial
Bare Soil
Cropland
Dense Forest
Grassland
Low Density Urban
Low/Sparse Vegetation
Water
Brightness temperature (°C)
0
5
10
15
20
25
30
35
40
45
50
28/4/13 28/6/13 28/8/13 28/10/13 28/12/13 28/2/14 30/4/14 30/6/14 31/8/14 31/10/14 31/12/14 28/2/15 30/4/15 30/6/15 31/8/15 31/10/15 31/12/15 29/2/16 30/4/16 30/6/16 31/8/16 31/10/16 31/12/16 28/2/17 30/4/17 30/6/17 31/8/17 31/10/17 31/12/17 28/2/18 30/4/18 30/6/18 31/8/18 31/10/18 31/12/18 28/2/19 30/4/19 30/6/19 31/8/19 31/10/19 31/12/19 29/2/20 30/4/20 30/6/20 31/8/20 31/10/20 31/12/20 28/2/21 30/4/21 30/6/21 31/8/21 31/10/21 31/12/21 28/2/22 30/4/22 30/6/22 31/8/22 31/10/22
Acquisition date



Band 11
Artificial
Bare Soil
Cropland
Dense Forest
Grassland
Low Density Urban
Low/Sparse Vegetation
Water
Brightness temperature (°C)
0
5
10
15
20
25
30
35
40
45
28/4/13 28/6/13 28/8/13 28/10/13 28/12/13 28/2/14 30/4/14 30/6/14 31/8/14 31/10/14 31/12/14 28/2/15 30/4/15 30/6/15 31/8/15 31/10/15 31/12/15 29/2/16 30/4/16 30/6/16 31/8/16 31/10/16 31/12/16 28/2/17 30/4/17 30/6/17 31/8/17 31/10/17 31/12/17 28/2/18 30/4/18 30/6/18 31/8/18 31/10/18 31/12/18 28/2/19 30/4/19 30/6/19 31/8/19 31/10/19 31/12/19 29/2/20 30/4/20 30/6/20 31/8/20 31/10/20 31/12/20 28/2/21 30/4/21 30/6/21 31/8/21 31/10/21 31/12/21 28/2/22 30/4/22 30/6/22 31/8/22 31/10/22
Acquisition date

Annex C - Python scripts

Step 1: Unzipping the Landsat scenes and performing the atmospheric correction

```
import os, subprocess, pathlib, sys
import tarfile, numpy
from osgeo import gdal


def imread(image):
      img = gdal.Open(image)
      im_array = numpy.array(img.ReadAsArray())
      return numpy.uint16(im_array), img.GetProjection(),
img.GetGeoTransform()


def imwrite (fileName, frmt, projection, geotransform, data) :
    drv = gdal.GetDriverByName(frmt)
    rows = data.shape[1]
    cols = data.shape[0]
    out = drv.Create(fileName, rows, cols, 1, gdal.GDT_Float64)
    band = out.GetRasterBand(1)
    band.WriteArray(data)
    band = None
    out.SetProjection(projection)
    out.SetGeoTransform(geotransform)
    out = None


def extract(path,tilelist):
    tilenamelist = []
    for tile in tilelist:
        tf = tarfile.open(os.path.join(path,tile))
        extraction_path=os.path.join(path,tile[:-7])
        if pathlib.Path(extraction_path).exists()==False:
            pathlib.Path(extraction_path).mkdir(parents=True)
            os.chdir(extraction_path)
            tf.extractall()
        tilenamelist.append(tile[:-7])
    return tilenamelist


def delete_tarfiles(path):
    [tars.append(i for i in os.listdir(path) if i.endswith('gz'))]
    for i in tars:
        os.remove(os.path.join(path,i))
    return 'tar files deleted successfully'


def conversion_decimal(string):
    if string[-1]=='2':
        number = float(string[:-4])*0.01
    elif string[-1]=='3':
        number = float(string[:-4])*0.001
```

```
    elif string[-1]=='4':
        number = float(string[:-4])*0.0001
    elif string[-1]=='5':
        number = float(string[:-4])*0.00001
    else:
        print('Error in MTL. Exiting processing')
        sys.exit()
    return number

def parseMTL(path):
    fl = open(path)
    metadata = {}
    for row in fl:
        if "=" in row:
            dt = row.split("=")
            metadata[dt[0].replace(" ","")] = dt[1].replace("\n","")
    return metadata


def atmcorr_landsat(path,tile):
    MLi = 'RADIANCE_MULT_BAND_'
    ALi = 'RADIANCE_ADD_BAND_'
    Mi = 'REFLECTANCE_MULT_BAND_'
    Ai = 'REFLECTANCE_ADD_BAND_'
    SE = 'SUN_ELEVATION'
    K1i = 'K1_CONSTANT_BAND_'
    K2i = 'K2_CONSTANT_BAND_'
    current_folder = os.path.join(path,tile)
    mtlFile = os.path.join(current_folder, tile + '_T1_MTL.txt')
    metaData = parseMTL(mtlFile)

    se = float(metaData[SE])

    for band in [1,2,3,4,5,6,7,8,9]:
        M = Mi+'{0}'.format(band)
        A = Ai+'{0}'.format(band)

        if M not in metaData or A not in metaData:
            continue

        M_val = conversion_decimal(metaData[M])
        A_val = float(metaData[A])

        image = [i for i in os.listdir(current_folder) if
i.endswith('B{0}.TIF'.format(band))]
        img = imread(os.path.join(current_folder,image[0]))

        spectral_reflectance_band = M_val*img[0]+A_val
        toa_reflectance_band = spectral_reflectance_band/numpy.sin(se *
numpy.pi/180.)
        toa_reflectance_band = numpy.where(img[0]==0,0,toa_reflectance_band)
        filename =
os.path.join(current_folder,tile+'_B{0}_refl.TIF'.format(band))
        imwrite (filename, 'GTiff', img[1], img[2], toa_reflectance_band)

    for band in [10, 11]:
        ML = MLi+'{0}'.format(band)
```

```
        AL = ALi+'{0}'.format(band)
        K1 = K1i+'{0}'.format(band)
        K2 = K2i+'{0}'.format(band)

        if ML not in metaData or AL not in metaData or K1 not in metaData or 
K2 not in metaData:
            continue

        ML_val = conversion_decimal(metaData[ML])
        AL_val = float(metaData[AL])
        K1_val = float(metaData[K1])
        K2_val = float(metaData[K2])

        image = [i for i in os.listdir(current_folder) if 
i.endswith('B{0}.TIF'.format(band))]
        img = imread(os.path.join(current_folder,image[0]))

        spectral_radiance_band = ML_val*img[0]+AL_val
        toa_brightness_temperature = 
K2_val/numpy.log((K1_val/spectral_radiance_band)+1)-273.
        toa_brightness_temperature = 
numpy.where(img[0]==0,0,toa_brightness_temperature)
        filename = 
os.path.join(current_folder,tile+'_B{0}_temp.TIF'.format(band))
        imwrite (filename, 'GTiff', img[1], img[2], 
toa_brightness_temperature)

#==============================================
PATH = "/path/to/imagery/folder/"
years = [ '2013','2014', '2015', '2016','2017','2018', '2019', '2020', 
'2021','2022' ]

for year in years:
    path_process=os.path.join(PATH,year)
    print(path_process)

    tarBalls = [f for f in os.listdir(path_process) if f.endswith(".tar")]

    subfolders = extract(path_process,tarBalls)

    for scene in subfolders:
        atmcorr_landsat(path_process,scene)
```

Step 2: Clipping Landsat scenes to the extents of the area of interest

```
import os, sys


#==============================================
PATH = '/path/to/imagery/folder/'
years = ['2013', '2014','2015','2016','2017','2018','2019', '2020', '2021', 
'2022']
aoi_path = '/path/to/AOI/extent/shapefile/'

for year in years:
    path_process=os.path.join(PATH,year)
```

```
    print(path_process)

    subfolders = [f for f in os.listdir(path_process) if not
f.endswith(".tar")]

    for scene in subfolders:
        imagelist=[]
        current_path1 = os.path.join(path_process,scene)
        print(current_path1)
        for i in os.listdir(current_path1):
            if i.endswith('refl.TIF') or i.endswith('temp.TIF'):
                imagelist.append(i)
        for image in imagelist:
            current_path2 = os.path.join(current_path1,image)
            os.system('gdalwarp -srcnodata 0 -overwrite -crop_to_cutline -
cutline {0} {1} {2}'.format(aoi_path,current_path2,current_path2[:-
4]+'_clip.tif'))
```

Step 3: Creating pixel-based samples from polygons

```
import os, math, numpy

import numpy as np
from osgeo import ogr, gdal, osr
from AlignToGrid import AlignToGrid

def geomRasterizer(id, geom, refGrid, resDict, resolution, parcEPSG=32634,
destEPSG=32634):
    drv = ogr.GetDriverByName('MEMORY')
    ds = drv.CreateDataSource("tmp")
    parcOSR = osr.SpatialReference()
    parcOSR.ImportFromEPSG(int(parcEPSG))

    destOSR = osr.SpatialReference()
    destOSR.ImportFromEPSG(int(destEPSG))

    lr = ds.CreateLayer("tmpftlr",parcOSR)
    idField = ogr.FieldDefn("id", ogr.OFTInteger64)
    lr.CreateField(idField)

    #create new ogr feature
    parc = ogr.Feature(lr.GetLayerDefn())
    parc.SetField("id",id)
    parc.SetGeometry(geom)
    allignedGrid = AlignToGrid(parc, refGrid)
    grd = allignedGrid.process(vector=True)
    drv = gdal.GetDriverByName("MEM")

    tmpDataset =  drv.Create("__del\\{0}.tif".format(parc.GetField("id")),
    int((grd[1][0] - grd[0][0]) / resolution), int((grd[0][1] - grd[1][1]) /
resolution), 1, gdal.GDT_Byte)

tmpDataset.SetProjection(parc.GetGeometryRef().GetSpatialReference().ExportTo
Wkt())
```

```
    tmpDataset.SetGeoTransform((grd[0][0], resolution, 0, grd[0][1], 0, -
resolution))

    # create temporary dataset
    ogrDrv = ogr.GetDriverByName("MEMORY")
    memVSource = ogrDrv.CreateDataSource(str(parc.GetField("id")))
    memVLayer = memVSource.CreateLayer("tmp", destOSR,
geom_type=ogr.wkbPolygon)
    tmpFtDefn = memVLayer.GetLayerDefn()
    ft = ogr.Feature(tmpFtDefn)
    ft.SetGeometry(parc.geometry())
    memVLayer.CreateFeature(ft)
    gdal.RasterizeLayer(tmpDataset, [1], memVLayer, burn_values=[1, ])
    resDict[parc.GetField("id")] = {"gt":tmpDataset.GetGeoTransform(),
"prj":tmpDataset.GetProjection(),
        "mask":tmpDataset.ReadAsArray(), "RasterXSize":
tmpDataset.RasterXSize,
        "RasterYSize":tmpDataset.RasterYSize}
    tmpDataset = None

def imBlockRead(path, res, id_):
    tmpDt = gdal.Open(path)
    tmpGt = tmpDt.GetGeoTransform()
    col, row = xyToRowCol(res[id_]["gt"][0], res[id_]["gt"][3], tmpGt)
    tmpArray = tmpDt.GetRasterBand(1).ReadAsArray(col, row,
res[id_]["RasterXSize"], res[id_]["RasterYSize"])
    if tmpArray is None:
        return None

    tmpArray = tmpArray.astype(float)
    tmpArray[res[id_]["mask"] == 0] = numpy.nan
    tmpArray[tmpArray == tmpDt.GetRasterBand(1).GetNoDataValue()] = numpy.nan
    return tmpArray



def imread(image):
      img = gdal.Open(image)
      im_array = numpy.array(img.ReadAsArray())
      return im_array, img.GetProjection(), img.GetGeoTransform()

def xyToRowCol(X, Y, gt):
    y = int((Y - gt[3]-gt[4]/gt[1]*X+gt[0]*gt[4]/gt[1])/(gt[5]-
(gt[2]*gt[4]/gt[1])))
    x = int((X-gt[0]-gt[2]*y)/gt[1])
    return [x,y]


    #=============================================
aoi_path = '/path/to/AOI/extents/'
reference_image =
"/path/to/reference/image/LC09_L1TP_185033_20220531_20220601_02/LC09_L1TP_185
033_20220531_20220601_02_B6_refl_clip.tif"

PATH = '/path/to/imagery/folder/'
DEM_PATH = '/path/to/dem_aligned.tif'
SLOPE_PATH = '/path/to/slope_aligned.tif'
```

```
TPI_PATH = '/path/to/tpi_aligned.tif'

aoi_path = '/polygon/samples/training_poly.gpkg'
shp_name = 'training_poly_utm'

samples_path = '/folder/for/the/pixelbased/samples/'

TEMP_fold = '/temporary/folder/'
im_path_out = os.path.join(TEMP_fold,'output.tif')
sample_path_out = os.path.join(TEMP_fold,'sample.shp')
dem_path_out = os.path.join(TEMP_fold,'dem.tif')
slope_path_out = os.path.join(TEMP_fold,'slope.tif')
tpi_path_out = os.path.join(TEMP_fold,'tpi.tif')
    #==============================================
years = ['2013', '2014','2015','2016','2017','2018','2019', '2020', '2021',
'2022']
names = ['year','month','day','poly_id', 'pixel_id',
'B2','B3','B4','B5','B6','B7','B10','B11','elevation','slope','tpi','class']

classes =
["artificial","bare_soil","cropland","dense_forest","low_density_urban","low_
sparse_vegetation","water"]

    #==============================================
txt_file = open(os.path.join(samples_path,'dataset.csv'),"w+")
txt_file.write(",".join(names))
txt_file.write("\n")
file = ogr.Open(aoi_path)
shape = file.GetLayer()
refImage = gdal.Open(reference_image)
gt = refImage.GetGeoTransform()
refImage = None

for feature in shape:
    cat = feature.GetField("class")
    id_ = feature.GetFID()
    print("Processing id: ", id_)
    res = {}
    geomRasterizer(id_, feature.geometry(), reference_image, res, gt[1])

    rawDt = [None]*11

    rawDt[-3] = imBlockRead(DEM_PATH, res, id_).flatten()
    rawDt[-2] = imBlockRead(SLOPE_PATH, res, id_).flatten()
    rawDt[-1] = imBlockRead(TPI_PATH, res, id_).flatten()



    for year in years:
        path_process=os.path.join(PATH,year)
        subfolders = [f for f in os.listdir(path_process) if not
f.endswith(".tar")]

        for scene in subfolders:
            current_path1 = os.path.join(path_process,scene)

            date = os.path.split(current_path1)[1].split("_")[3]
```

```
            bandId = 0
            for attr in names_test[4::]:
                for i in os.listdir(current_path1):
                    if i.endswith('.tif') and attr in i:
                        path = os.path.join(current_path1, i)
                        rawDt[bandId] = imBlockRead(path, res, id_).flatten()
                        bandId += 1

            rowOffset = 5
            pixelCount = rawDt[bandId].shape[0]

            for i in range(pixelCount):
                if np.isnan(rawDt[0][i]):
                    continue

                new_row = list(range(len(names_test)))
                new_row[0] = date[0:4]
                new_row[1] = date[4:6]
                new_row[2] = date[6:8]
                new_row[3] = str(id_)
                new_row[4] = str(i)
                isNone = False


                k = 0
                for bnd in rawDt:
                    new_row[rowOffset+k] = str(bnd[i])
                    k += 1

                new_row[16] = cat
                txt_file.write(', '.join(new_row) + '\n')

txt_file.close()
```

Step 4: Building of the time series dataset

```
import psycopg2, numpy as np

cnStr = "dbname=thesis user=postgres"
cn = psycopg2.connect(cnStr)


query = "SELECT DISTINCT year, month, day FROM dataset_cloud_free ORDER BY
year, month,day"
cursor = cn.cursor()
cursor.execute(query)
dates = cursor.fetchall()
dateCount = len(dates)
print(dateCount)

query = "SELECT DISTINCT  elevation, slope, tpi, poly_id, pixel_id, class
FROM cloudfree ORDER BY poly_id, pixel_id"
cursor = cn.cursor()
cursor.execute(query)
```

```
polyIDs = cursor.fetchall()
cols = ["b2", "b3", "b4", "b5", "b6", "b7", "b10", "b11"]
outFile = open("/dataset_cloud_free_timeseries.csv", "w")

header = []
for col in cols:
        for date in dates:
            header += [col+"({0}-{1}-{2})".format(*date)]

header+=["elevation", "slope", "tpi", "poly_id", "pixel_id", "class"]
outFile.write(",".join(header))
outFile.write("\n")


for rowDt in polyIDs:
    print(rowDt[-3], rowDt[-2])
    #reading multispectral info
    query = """SELECT {0}
        FROM cloudfree dv
        WHERE poly_id='{1}' and pixel_id = '{2}'
        ORDER BY YEAR,MONTH,day""".format(",".join(cols), rowDt[-3], rowDt[-
2])
    cursor = cn.cursor()
    cursor.execute(query)
    timeseries = cursor.fetchall()
    timeseries = np.array(timeseries).T.flatten()

    #appending terrain, class, and id info

    row = timeseries.tolist() + list(rowDt)

    outFile.write(",".join(row))
    outFile.write("\n")
outFile.close()
```

Step 5: kNearest Neighbor and Random Forests model training and classification

```
import numpy as np, os
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix,
ConfusionMatrixDisplay
from pandas import read_csv
import matplotlib.pyplot as plt
import seaborn as sn
from datetime import datetime
from multiprocessing import Process, Manager
from osgeo import gdal


def imread(image):
```

```
      img = gdal.Open(image)
      im_array = np.array(img.ReadAsArray())
      return im_array, img.GetProjection(), img.GetGeoTransform()

def imwrite (fileName, frmt, projection, geotransform, data) :
    drv = gdal.GetDriverByName(frmt)
    rows = data.shape[1]
    cols = data.shape[0]
    out = drv.Create(fileName, rows, cols, 1, gdal.GDT_Float32)
    band = out.GetRasterBand(1)
    band.WriteArray(data)
    band = None
    out.SetProjection(projection)
    out.SetGeoTransform(geotransform)
    out = None

def chunkIt(seq, num):
    avg = len(seq) / float(num)
    out = []
    last = 0.0

    while last < len(seq):
        out.append(seq[int(last):int(last + avg)])
        last += avg

    return out


class ComputeModels():

    def __del__(self):
        self.log.close()
        self.log = None

    def __init__(self, dataPath, outPath, samplesFile, cols2use, mode,
seasonDivision=0):

        self.PATH_in = dataPath
        self.classification_output_path = outPath
        os.makedirs(self.classification_output_path, exist_ok=True)

        self.classes =
{"artificial":0,"bare_soil":1,"cropland":2,"dense_forest":3,
"grassland":4,"low_density_urban":5,
                        "low_sparse_vegetation":6,"water":7}
        self.classNames = ["artificial", "bare_soil", "cropland",
"dense_forest", "grassland", "low_density_urban",
                           "low_sparse_vegetation", "water"]
        self.samplesFile = samplesFile
        self.cols2use = cols2use
        self.mode = mode
        logFile = 'log_'+'_' + str(mode) +'.txt'

        self.log = open(os.path.join(self.classification_output_path,
logFile),"w+")
```

```
        samples = open(self.samplesFile, "r")
        self.dataset = read_csv(samples,low_memory=False)
        seasonData = {'all': [[], []]}
        if seasonDivision == 1:
            seasonData = 
{'all':[[],[]],'summer':[[],[]],'autumn':[[],[]],'winter':[[],[]],'spring':[[
],[]]}
        tmpVals = self.dataset.values
        selectedVals = self.dataset[cols2use].values
        tmpLabels = [x.replace(" ","") for x in tmpVals[:,-1]]
        if seasonDivision == 1:
            for j in range (0, tmpVals.shape[0]):

                key = None
                if tmpVals[j][1] == 12 or tmpVals[j][1] == 1 or tmpVals[j][1] 
== 2:
                    key = "winter"
                elif tmpVals[j][1] == 3 or tmpVals[j][1] == 4 or 
tmpVals[j][1] == 5:
                    key = "spring"
                elif tmpVals[j][1] == 6 or tmpVals[j][1] == 7 or 
tmpVals[j][1] == 8:
                    key = "summer"
                elif tmpVals[j][1] == 9 or tmpVals[j][1] == 10 or 
tmpVals[j][1] == 11:
                    key = "autumn"

                seasonData[key][0].append(selectedVals[j])
                seasonData[key][1].append(self.classes[tmpLabels[j] ])

        seasonData["all"][0] = selectedVals
        seasonData["all"][1] = [self.classes[x] for x in tmpLabels]

        # Split-out validation dataset
        self.trainXMin = {}
        self.trainXMax = {}
        self.X_train = {}
        self.X_validation = {}
        self.Y_train = {}
        self.Y_validation = {}
        for season in seasonData:
            seasonData[season][0] = np.array(seasonData[season][0])
            seasonData[season][1] = np.array(seasonData[season][1]).reshape(-
1,1)
            self.trainXMin[season] = seasonData[season][0].min(axis = 0)
            self.trainXMax[season] = seasonData[season][0].max(axis = 0)

            self.X_train[season], self.X_validation[season], 
self.Y_train[season], self.Y_validation[season] = 
train_test_split(seasonData[season][0], seasonData[season][1], 
test_size=0.20, random_state=1, shuffle=True)
            self.Y_train[season] = self.Y_train[season].flatten()
            self.Y_validation[season] = self.Y_validation[season].flatten()


    def trainKNeighbors(self):
        self.KNmodel = {}
```

```
        for season in self.X_train:
            self.KNmodel[season] = KNeighborsClassifier()
            self.KNmodel[season].fit(self.X_train[season],
self.Y_train[season])

    def trainRandomForest(self):
        self.RandomForestModel = {}

        for season in self.X_train:
            self.RandomForestModel[season] = RandomForestClassifier()
            self.RandomForestModel[season].fit(self.X_train[season],
self.Y_train[season])


    def predictModel(self, model, xval, yval):

        for season in xval:
            self.log.write('\n\n{} prediction results for season:
{}'.format(model[season],season) + '\n')
            model[season].n_jobs = 24
            predictions = model[season].predict(xval[season])
            self.log.write(str(accuracy_score(yval[season], predictions)) +
'\n')
            self.log.write(str(confusion_matrix(yval[season], predictions)) +
'\n')
            self.log.write(str(classification_report(yval[season],
predictions)) + '\n')


            showClasses = [self.classNames[i].replace("_"," ") for i in
model[season].classes_]



            cm = confusion_matrix(yval[season], predictions)
            cm = cm/ cm.sum(axis=1)
            cm = np.round(cm,3)
            fig, ax = plt.subplots(figsize=(20, 20))
            ax.matshow(cm, cmap=plt.cm.Blues, alpha=0.6)
            plt.xlabel('Predictions', fontsize=18)
            plt.ylabel('Reference', fontsize=18)
            plt.title('Confusion Matrix for season: {0}'.format(season),
fontsize=23)
            for i in range(cm.shape[0]):
                for j in range(cm.shape[1]):
                    ax.text(x=j, y=i, s=cm[i, j], va='center', ha='center',
size='xx-large')

            ax.set_yticks(list(range(len(showClasses))), showClasses,
fontsize=15, rotation=30)
            ax.set_xticks(list(range(len(showClasses))), showClasses,
fontsize=15, rotation=20)
            plt.subplots_adjust(top=0.88)


plt.savefig(os.path.join(self.classification_output_path,'confusion_matrix_{0
}.jpg'.format(season)))
```

```
            plt.close()



    def writeClassificationResult(self, model, modelName, demPath=None,
slopePath=None, tpiPath=None):

        years = ['2018',] #'2014','2015','2016','2017','2018','2019'
        for year in years:
            path_process=os.path.join(self.PATH_in,year)
            print(path_process)

            subfolders = [f for f in os.listdir(path_process) if not
f.endswith(".tar")]

            for scene in subfolders:
                inData = os.path.join(path_process,scene)
                pathRow = scene.split("_")[2]
                inListFiles = os.listdir(inData)
                inListArray = []
                projection = None
                geoTransform = None
                for band in self.cols2use:
                    for fileName in inListFiles:
                        if band.upper() in fileName and
fileName.endswith(".tif") and "clip" in fileName:
                            tmpDataset = gdal.Open(os.path.join(inData,
fileName))
                            inListArray.append(tmpDataset.ReadAsArray())
                            projection = tmpDataset.GetProjection()
                            geoTransform = tmpDataset.GetGeoTransform()

                if "elevation" in self.cols2use:
                    demImage, demProjection, demGeoTransform =
imread(demPath)
                    inListArray.append(demImage)


                if "slope" in self.cols2use:
                    slopeImage, slopeProjection, slopeGeoTransform =
imread(slopePath)
                    inListArray.append(slopeImage)

                if "tpi" in self.cols2use:
                    tpiImage = imread(os.path.join(tpiPath))[0]
                    inListArray.append(tpiImage)

                inDataset = np.array(inListArray)
                normalizedDataset =
inDataset.T.reshape(inDataset.shape[1]*inDataset.shape[2],
inDataset.shape[0])
                model["all"].n_jobs = 8

                outPath = os.path.join(self.classification_output_path,year)
                os.makedirs(outPath, exist_ok=True)

                outBand = model["all"].predict(normalizedDataset)
```

```
                imwrite(os.path.join(outPath,scene
+'_'+'_'.join(self.cols2use)+'_{0}_class.tif'.format(modelName)), "GTiff",
projection, geoTransform, outBand.reshape((inDataset.shape[2],
inDataset.shape[1])).T)
                print("ok!")
                return


def writeClassificationResultTimeseries(self, model, modelName, dates,
uniqueBands, demPath=None, slopePath=None, tpiPath=None ):
        spectralBands = uniqueBands
        if "elevation" in uniqueBands:
            spectralBands = uniqueBands[0:-3]
        bandFiles = []

        for band in spectralBands:
            for date in dates:
                mergeDate = str(date[0])+date[1].replace("
","")+date[2].replace(" ","")
                dataPath = os.path.join(self.PATH_in, str(date[0]))
                dataset = [f for f in os.listdir(dataPath) if not
f.endswith(".tar") and mergeDate == f.split("_")[3]][0]
                dataPath = os.path.join(dataPath, dataset)
                file = [f for f in os.listdir(dataPath) if "clip" in f and
band.upper() in f and ".xml" not in f][0]
                bandFiles.append(os.path.join(dataPath, file))
                #bandFiles.append(file)

        #appending elevation data
        if "elevation" in uniqueBands:
            bandFiles.append(demPath)
            bandFiles.append(slopePath)
            bandFiles.append(tpiPath)

        #reading reference image
        #tmpDt = gdal.Open(bandFiles[0])
        sampleFile = gdal.Open(bandFiles[0])
        dims = [sampleFile.RasterXSize,sampleFile.RasterYSize]

        drv = gdal.GetDriverByName("GTiff")
        outFile = os.path.join(self.classification_output_path, "output.tif")
        outDataset = drv.Create(outFile, dims[0], dims[1], 1,
gdal.GDT_Float32)

        gt = list(sampleFile.GetGeoTransform())

        outDataset.SetGeoTransform(gt)
        outDataset.SetProjection(sampleFile.GetProjection())

        outDataset = None
        sampleFile = None
        nThreads = 8
        chunks = chunkIt(range(dims[1]), nThreads)
        threads = list(range(nThreads))

        for thread in range(nThreads):
```

```
            print(chunks[thread])
            threads[thread] = Process(target=processRegion, args=(bandFiles,
chunks[thread], dims, model, outFile))
            threads[thread].start()

        for trd in threads:
            trd.join()

        return 0




def main():
    filePath = '/path/to/imagery/folder/'
    outPath = "/output/path/of/results/"
    dataset = "/path/to/samples/dataset_cloud_free.csv"
    demPath = '/path/to/dem_aligned.tif'
    slopePath = '/path/to/slope_aligned.tif'
    tpiPath = '/path/to/tpi_aligned.tif'

    trainingModes = {

        "multispectral": {
            "columns_to_use": ['b2', 'b3', 'b4', 'b5', 'b6', 'b7']
        },
        "multispectral_thermal": {
            "columns_to_use": ['b2', 'b3', 'b4', 'b5', 'b6', 'b7', "b10",
"b11"]
        },
        "multispectral_terrain": {
            "columns_to_use": ['b2', 'b3', 'b4', 'b5', 'b6',
'b7','elevation','slope','tpi']
        },
        "multispectral_thermal_terrain": {
            "columns_to_use":['b2', 'b3', 'b4', 'b5', 'b6', 'b7',"b10",
"b11",'elevation','slope','tpi']
        }
    }
    for algorithm in ["RF", "kNN", ]:
        print("Algorithm: ", algorithm)
        for mode in trainingModes:
            print("Performing mode: ", mode)
            a = ComputeModels(filePath, os.path.join(outPath,*[mode,
algorithm]) , dataset,

trainingModes[mode]["columns_to_use"], mode, 0)
            model=None
            if(algorithm == "kNN"):
                a.trainKNeighbors()
                model = a.KNmodel
            elif(algorithm == "RF"):
                a.trainRandomForest()
                model = a.RandomForestModel

            a.predictModel(model, a.X_validation, a.Y_validation)
            a.writeClassificationResult(model,algorithm, demPath, slopePath,
```

```
tpiPath)
            #computing classification results


if __name__ == "__main__":
    main()
```

Step 6: Random Forests time series model training and classification

```
import psycopg2, os
from train_classify_v2 import ComputeModels


def main():
    filePath = '/path/to/imagery/folder/'
    dataset = "/path/to/samples/dataset_cloud_free_timeseries.csv"
    outPath = "/output/path/for/results"
    demPath = '/path/to/dem_aligned.tif'
    slopePath = '/path/to/slope_aligned.tif'
    tpiPath = '/path/to/tpi_aligned.tif'

    cnStr = "dbname=thesis user=postgres"
    cn = psycopg2.connect(cnStr)
    query = "SELECT DISTINCT '('||year || '-' || month || '-' || day || ')',
year, month, day FROM dataset_cloud_free ORDER BY year, month, day"
    cursor = cn.cursor()
    cursor.execute(query)
    dates = cursor.fetchall()

    terrainCols = ['elevation','slope','tpi']

    trainingModes = {
        "multispectral": {
            "columns_to_use": ['b2', 'b3', 'b4', 'b5', 'b6', 'b7']
        },
        "multispectral_thermal": {
            "columns_to_use": ['b2', 'b3', 'b4', 'b5', 'b6', 'b7', "b10",
"b11"]
        },

        "multispectral_terrain": {
            "columns_to_use": ['b2', 'b3', 'b4', 'b5', 'b6',
'b7']+terrainCols
        },
        "multispectral_thermal_terrain": {
            "columns_to_use":['b2', 'b3', 'b4', 'b5', 'b6', 'b7',"b10",
"b11"]+terrainCols
        }
    }


    for mode in trainingModes:
        requestCols = []
```

```
        parseDate = []
        appendDate = True
        for col in trainingModes[mode]["columns_to_use"]:
            if col not in terrainCols:
                for row in dates:
                    requestCols.append(col+row[0])
                    if(appendDate):
                        parseDate.append(row[1:4])
                appendDate = False

        for col in terrainCols:
            if col in trainingModes[mode]["columns_to_use"]:
                requestCols.append(col)


        algorithm = "RF"
        a = ComputeModels(filePath, os.path.join(outPath, *[mode,
algorithm]), dataset, requestCols, mode, 0)
        a.trainRandomForest()
        model = a.RandomForestModel
        a.predictModel(model, a.X_validation, a.Y_validation)
        a.writeClassificationResultTimeseries(model,algorithm, parseDate,
trainingModes[mode]["columns_to_use"], demPath, slopePath, tpiPath)


    return 0




if __name__ == "__main__":
    main()
```

Series from Lund University

Department of Physical Geography and Ecosystem Science

Master Thesis in Geographical Information Science

1. *Anthony Lawther:* The application of GIS-based binary logistic regression for slope failure susceptibility mapping in the Western Grampian Mountains, Scotland (2008).
2. *Rickard Hansen:* Daily mobility in Grenoble Metropolitan Region, France. Applied GIS methods in time geographical research (2008).
3. *Emil Bayramov:* Environmental monitoring of bio-restoration activities using GIS and Remote Sensing (2009).
4. *Rafael Villarreal Pacheco:* Applications of Geographic Information Systems as an analytical and visualization tool for mass real estate valuation: a case study of Fontibon District, Bogota, Columbia (2009).
5. *Siri Oestreich Waage:* a case study of route solving for oversized transport: The use of GIS functionalities in transport of transformers, as part of maintaining a reliable power infrastructure (2010).
6. *Edgar Pimiento:* Shallow landslide susceptibility – Modelling and validation (2010).
7. *Martina Schäfer:* Near real-time mapping of floodwater mosquito breeding sites using aerial photographs (2010).
8. *August Pieter van Waarden-Nagel:* Land use evaluation to assess the outcome of the programme of rehabilitation measures for the river Rhine in the Netherlands (2010).
9. *Samira Muhammad:* Development and implementation of air quality data mart for Ontario, Canada: A case study of air quality in Ontario using OLAP tool. (2010).
10. *Fredros Oketch Okumu*: Using remotely sensed data to explore spatial and temporal relationships between photosynthetic productivity of vegetation and malaria transmission intensities in selected parts of Africa (2011).

11. *Svajunas Plunge:* Advanced decision support methods for solving diffuse water pollution problems (2011).

12. *Jonathan Higgins:* Monitoring urban growth in greater Lagos: A case study using GIS to monitor the urban growth of Lagos 1990 - 2008 and produce future growth prospects for the city (2011).

13. *Mårten Karlberg:* Mobile Map Client API: Design and Implementation for Android (2011).

14. *Jeanette McBride:* Mapping Chicago area urban tree canopy using color infrared imagery (2011).

15. *Andrew Farina:* Exploring the relationship between land surface temperature and vegetation abundance for urban heat island mitigation in Seville, Spain (2011).

16. *David Kanyari*: Nairobi City Journey Planner: An online and a Mobile Application (2011).

17. *Laura V. Drews:* Multi-criteria GIS analysis for siting of small wind power plants - A case study from Berlin (2012).

18. *Qaisar Nadeem:* Best living neighborhood in the city - A GIS based multi criteria evaluation of ArRiyadh City (2012).

19. *Ahmed Mohamed El Saeid Mustafa:* Development of a photo voltaic building rooftop integration analysis tool for GIS for Dokki District, Cairo, Egypt (2012).

20. *Daniel Patrick Taylor*: Eastern Oyster Aquaculture: Estuarine Remediation via Site Suitability and Spatially Explicit Carrying Capacity Modeling in Virginia's Chesapeake Bay (2013).

21. *Angeleta Oveta Wilson:* A Participatory GIS approach to *unearthing* Manchester's Cultural Heritage '*gold mine'* (2013).

22. *Ola Svensson:* Visibility and Tholos Tombs in the Messenian Landscape: A Comparative Case Study of the Pylian Hinterlands and the Soulima Valley (2013).

23. *Monika Ogden:* Land use impact on water quality in two river systems in South Africa (2013).

24. *Stefan Rova:* A GIS based approach assessing phosphorus load impact on Lake Flaten in Salem, Sweden (2013).

25. *Yann Buhot:* Analysis of the history of landscape changes over a period of 200 years. How can we predict past landscape pattern scenario and the impact on habitat diversity? (2013).

26. *Christina Fotiou:* Evaluating habitat suitability and spectral heterogeneity models to predict weed species presence (2014).

27. *Inese Linuza:* Accuracy Assessment in Glacier Change Analysis (2014).

28. *Agnieszka Griffin:* Domestic energy consumption and social living standards: a GIS analysis within the Greater London Authority area (2014).

29. *Brynja Guðmundsdóttir:* Detection of potential arable land with remote sensing and GIS - A Case Study for Kjósarhreppur (2014).

30. *Oleksandr Nekrasov:* Processing of MODIS Vegetation Indices for analysis of agricultural droughts in the southern Ukraine between the years 2000-2012 (2014).

31. *Sarah Tressel:* Recommendations for a polar Earth science portal in the context of Arctic Spatial Data Infrastructure (2014).

32. *Caroline Gevaert:* Combining Hyperspectral UAV and Multispectral Formosat-2 Imagery for Precision Agriculture Applications (2014).

33. *Salem Jamal-Uddeen:* Using GeoTools to implement the multi-criteria evaluation analysis - weighted linear combination model (2014).

34. *Samanah Seyedi-Shandiz:* Schematic representation of geographical railway network at the Swedish Transport Administration (2014).

35. *Kazi Masel Ullah:* Urban Land-use planning using Geographical Information System and analytical hierarchy process: case study Dhaka City (2014).

36. *Alexia Chang-Wailing Spitteler:* Development of a web application based on MCDA and GIS for the decision support of river and floodplain rehabilitation projects (2014).

37. *Alessandro De Martino:* Geographic accessibility analysis and evaluation of potential changes to the public transportation system in the City of Milan (2014).

38. *Alireza Mollasalehi:* GIS Based Modelling for Fuel Reduction Using Controlled Burn in Australia. Case Study: Logan City, QLD (2015).

39. *Negin A. Sanati:* Chronic Kidney Disease Mortality in Costa Rica; Geographical Distribution, Spatial Analysis and Non-traditional Risk Factors (2015).

40. *Karen McIntyre:* Benthic mapping of the Bluefields Bay fish sanctuary, Jamaica (2015).

41. *Kees van Duijvendijk:* Feasibility of a low-cost weather sensor network for agricultural purposes: A preliminary assessment (2015).

42. *Sebastian Andersson Hylander:* Evaluation of cultural ecosystem services using GIS (2015).

43. *Deborah Bowyer:* Measuring Urban Growth, Urban Form and Accessibility as Indicators of Urban Sprawl in Hamilton, New Zealand (2015).

44. *Stefan Arvidsson:* Relationship between tree species composition and phenology extracted from satellite data in Swedish forests (2015).

45. *Damián Giménez Cruz*: GIS-based optimal localisation of beekeeping in rural Kenya (2016).

46. *Alejandra Narváez Vallejo:* Can the introduction of the topographic indices in LPJ-GUESS improve the spatial representation of environmental variables? (2016).

47. *Anna Lundgren:* Development of a method for mapping the highest coastline in Sweden using breaklines extracted from high resolution digital elevation models (2016).

48. *Oluwatomi Esther Adejoro:* Does location also matter? A spatial analysis of social achievements of young South Australians (2016).

49. *Hristo Dobrev Tomov:* Automated temporal NDVI analysis over the Middle East for the period 1982 - 2010 (2016).

50. *Vincent Muller:* Impact of Security Context on Mobile Clinic Activities A GIS Multi Criteria Evaluation based on an MSF Humanitarian Mission in Cameroon (2016).

51. *Gezahagn Negash Seboka:* Spatial Assessment of NDVI as an Indicator of Desertification in Ethiopia using Remote Sensing and GIS (2016).

52. *Holly Buhler:* Evaluation of Interfacility Medical Transport Journey Times in Southeastern British Columbia. (2016).

53. *Lars Ole Grottenberg*: Assessing the ability to share spatial data between emergency management organisations in the High North (2016).

54. *Sean Grant:* The Right Tree in the Right Place: Using GIS to Maximize the Net Benefits from Urban Forests (2016).

55. *Irshad Jamal:* Multi-Criteria GIS Analysis for School Site Selection in Gorno-Badakhshan Autonomous Oblast, Tajikistan (2016).

56. *Fulgencio Sanmartín:* Wisdom-volkano: A novel tool based on open GIS and time-series visualization to analyse and share volcanic data (2016).

57. *Nezha Acil:* Remote sensing-based monitoring of snow cover dynamics and its influence on vegetation growth in the Middle Atlas Mountains (2016).

58. *Julia Hjalmarsson:* A Weighty Issue: Estimation of Fire Size with Geographically Weighted Logistic Regression (2016).

59. *Mathewos Tamiru Amato:* Using multi-criteria evaluation and GIS for chronic food and nutrition insecurity indicators analysis in Ethiopia (2016).

60. *Karim Alaa El Din Mohamed Soliman El Attar:* Bicycling Suitability in Downtown, Cairo, Egypt (2016).

61. *Gilbert Akol Echelai:* Asset Management: Integrating GIS as a Decision Support Tool in Meter Management in National Water and Sewerage Corporation (2016).

62. *Terje Slinning:* Analytic comparison of multibeam echo soundings (2016).

63. *Gréta Hlín Sveinsdóttir:* GIS-based MCDA for decision support: A framework for wind farm siting in Iceland (2017).

64. *Jonas Sjögren:* Consequences of a flood in Kristianstad, Sweden: A GIS-based analysis of impacts on important societal functions (2017).

65. *Nadine Raska:* 3D geologic subsurface modelling within the Mackenzie Plain, Northwest Territories, Canada (2017).

66. *Panagiotis Symeonidis*: Study of spatial and temporal variation of atmospheric optical parameters and their relation with PM 2.5 concentration over Europe using GIS technologies (2017).

67. *Michaela Bobeck:* A GIS-based Multi-Criteria Decision Analysis of Wind Farm Site Suitability in New South Wales, Australia, from a Sustainable Development Perspective (2017).

68. *Raghdaa Eissa*: Developing a GIS Model for the Assessment of Outdoor Recreational Facilities in New Cities Case Study: Tenth of Ramadan City, Egypt (2017).

69. *Zahra Khais Shahid*: Biofuel plantations and isoprene emissions in Svea and Götaland (2017).

70. *Mirza Amir Liaquat Baig*: Using geographical information systems in epidemiology: Mapping and analyzing occurrence of diarrhea in urban - residential area of Islamabad, Pakistan (2017).

71. *Joakim Jörwall*: Quantitative model of Present and Future well-being in the EU-28: A spatial Multi-Criteria Evaluation of socioeconomic and climatic comfort factors (2017).

72. *Elin Haettner*: Energy Poverty in the Dublin Region: Modelling Geographies of Risk (2017).

73. *Harry Eriksson*: Geochemistry of stream plants and its statistical relations to soil- and bedrock geology, slope directions and till geochemistry. A GIS-analysis of small catchments in northern Sweden (2017).

74. *Daniel Gardevärn:* PPGIS and Public meetings – An evaluation of public participation methods for urban planning (2017).

75. *Kim Friberg:* Sensitivity Analysis and Calibration of Multi Energy Balance Land Surface Model Parameters (2017).

76. *Viktor Svanerud:* Taking the bus to the park? A study of accessibility to green areas in Gothenburg through different modes of transport (2017).

77. *Lisa-Gaye Greene*: Deadly Designs: The Impact of Road Design on Road Crash Patterns along Jamaica's North Coast Highway (2017).

78. *Katarina Jemec Parker*: Spatial and temporal analysis of fecal indicator bacteria concentrations in beach water in San Diego, California (2017).

79. *Angela Kabiru*: An Exploratory Study of Middle Stone Age and Later Stone Age Site Locations in Kenya's Central Rift Valley Using Landscape Analysis: A GIS Approach (2017).

80. *Kristean Björkmann*: Subjective Well-Being and Environment: A GIS-Based Analysis (2018).

81. *Williams Erhunmonmen Ojo*: Measuring spatial accessibility to healthcare for people living with HIV-AIDS in southern Nigeria (2018).

82. *Daniel Assefa*: Developing Data Extraction and Dynamic Data Visualization (Styling) Modules for Web GIS Risk Assessment System (WGRAS). (2018).

83. *Adela Nistora*: Inundation scenarios in a changing climate: assessing potential impacts of sea-level rise on the coast of South-East England (2018).

84. *Marc Seliger*: Thirsty landscapes - Investigating growing irrigation water consumption and potential conservation measures within Utah's largest master-planned community: Daybreak (2018).

85. *Luka Jovičić*: Spatial Data Harmonisation in Regional Context in Accordance with INSPIRE Implementing Rules (2018).

86. *Christina Kourdounouli*: Analysis of Urban Ecosystem Condition Indicators for the Large Urban Zones and City Cores in EU (2018).

87. *Jeremy Azzopardi*: Effect of distance measures and feature representations on distance-based accessibility measures (2018).

88. *Patrick Kabatha*: An open source web GIS tool for analysis and visualization of elephant GPS telemetry data, alongside environmental and anthropogenic variables (2018).

89. *Richard Alphonce Giliba*: Effects of Climate Change on Potential Geographical Distribution of Prunus africana (African cherry) in the Eastern Arc Mountain Forests of Tanzania (2018).

90. *Eiður Kristinn Eiðsson*: Transformation and linking of authoritative multi-scale geodata for the Semantic Web: A case study of Swedish national building data sets (2018).

91. *Niamh Harty*: HOP!: a PGIS and citizen science approach to monitoring the condition of upland paths (2018).

92. *José Estuardo Jara Alvear*: Solar photovoltaic potential to complement hydropower in Ecuador: A GIS-based framework of analysis (2018).

93. *Brendan O'Neill*: Multicriteria Site Suitability for Algal Biofuel Production Facilities (2018).

94. *Roman Spataru*: Spatial-temporal GIS analysis in public health – a case study of polio disease (2018).

95. *Alicja Miodońska*: Assessing evolution of ice caps in Suðurland, Iceland, in years 1986 - 2014, using multispectral satellite imagery (2019).

96. *Dennis Lindell Schettini*: A Spatial Analysis of Homicide Crime's Distribution and Association with Deprivation in Stockholm Between 2010-2017 (2019).

97. *Damiano Vesentini*: The Po Delta Biosphere Reserve: Management challenges and priorities deriving from anthropogenic pressure and sea level rise (2019).

98. *Emilie Arnesten*: Impacts of future sea level rise and high water on roads, railways and environmental objects: a GIS analysis of the potential effects of increasing sea levels and highest projected high water in Scania, Sweden (2019).

99. *Syed Muhammad Amir Raza*: Comparison of geospatial support in RDF stores: Evaluation for ICOS Carbon Portal metadata (2019).

100. *Hemin Tofiq*: Investigating the accuracy of Digital Elevation Models from UAV images in areas with low contrast: A sandy beach as a case study (2019).

101. *Evangelos Vafeiadis*: Exploring the distribution of accessibility by public transport using spatial analysis. A case study for retail concentrations and public hospitals in Athens (2019).

102. *Milan Sekulic*: Multi-Criteria GIS modelling for optimal alignment of roadway by-passes in the Tlokweng Planning Area, Botswana (2019).

103. *Ingrid Piirisaar*: A multi-criteria GIS analysis for siting of utility-scale photovoltaic solar plants in county Kilkenny, Ireland (2019).

104. *Nigel Fox*: Plant phenology and climate change: possible effect on the onset of various wild plant species' first flowering day in the UK (2019).

105. *Gunnar Hesch*: Linking conflict events and cropland development in Afghanistan, 2001 to 2011, using MODIS land cover data and Uppsala Conflict Data Programme (2019).

106. *Elijah Njoku*: Analysis of spatial-temporal pattern of Land Surface Temperature (LST) due to NDVI and elevation in Ilorin, Nigeria (2019).

107. *Katalin Bunyevácz*: Development of a GIS methodology to evaluate informal urban green areas for inclusion in a community governance program (2019).

108. *Paul dos Santos*: Automating synthetic trip data generation for an agent-based simulation of urban mobility (2019).

109. *Robert O' Dwyer*: Land cover changes in Southern Sweden from the mid-Holocene to present day: Insights for ecosystem service assessments (2019).

110. *Daniel Klingmyr*: Global scale patterns and trends in tropospheric NO2 concentrations (2019).

111. *Marwa Farouk Elkabbany*: Sea Level Rise Vulnerability Assessment for Abu Dhabi, United Arab Emirates (2019).

112. *Jip Jan van Zoonen*: Aspects of Error Quantification and Evaluation in Digital Elevation Models for Glacier Surfaces (2020).

113. *Georgios Efthymiou*: The use of bicycles in a mid-sized city – benefits and obstacles identified using a questionnaire and GIS (2020).

114. *Haruna Olayiwola Jimoh*: Assessment of Urban Sprawl in MOWE/IBAFO Axis of Ogun State using GIS Capabilities (2020).

115. *Nikolaos Barmpas Zachariadis*: Development of an iOS, Augmented Reality for disaster management (2020).

116. *Ida Storm*: ICOS Atmospheric Stations: Spatial Characterization of CO2 Footprint Areas and Evaluating the Uncertainties of Modelled CO2 Concentrations (2020).

117. *Alon Zuta*: Evaluation of water stress mapping methods in vineyards using airborne thermal imaging (2020).

118. *Marcus Eriksson*: Evaluating structural landscape development in the municipality Upplands-Bro, using landscape metrics indices (2020).

119. *Ane Rahbek Vierø*: Connectivity for Cyclists? A Network Analysis of Copenhagen's Bike Lanes (2020).

120. *Cecilia Baggini*: Changes in habitat suitability for three declining Anatidae species in saltmarshes on the Mersey estuary, North-West England (2020).

121. *Bakrad Balabanian*: Transportation and Its Effect on Student Performance (2020).

122. *Ali Al Farid*: Knowledge and Data Driven Approaches for Hydrocarbon Microseepage Characterizations: An Application of Satellite Remote Sensing (2020).

123. *Bartlomiej Kolodziejczyk*: Distribution Modelling of Gene Drive-Modified Mosquitoes and Their Effects on Wild Populations (2020).

124. *Alexis Cazorla*: Decreasing organic nitrogen concentrations in European water bodies - links to organic carbon trends and land cover (2020).

125. *Kharid Mwakoba*: Remote sensing analysis of land cover/use conditions of community-based wildlife conservation areas in Tanzania (2021).

126. *Chinatsu Endo*: Remote Sensing Based Pre-Season Yellow Rust Early Warning in Oromia, Ethiopia (2021).

127. *Berit Mohr*: Using remote sensing and land abandonment as a proxy for long-term human out-migration. A Case Study: Al-Hassakeh Governorate, Syria (2021).

128. *Kanchana Nirmali Bandaranayake*: Considering future precipitation in delineation locations for water storage systems - Case study Sri Lanka (2021).

129. *Emma Bylund*: Dynamics of net primary production and food availability in the aftermath of the 2004 and 2007 desert locust outbreaks in Niger and Yemen (2021).

130. *Shawn Pace*: Urban infrastructure inundation risk from permanent sea-level rise scenarios in London (UK), Bangkok (Thailand) and Mumbai (India): A comparative analysis (2021).

131. *Oskar Evert Johansson*: The hydrodynamic impacts of Estuarine Oyster reefs, and the application of drone technology to this study (2021).

132. *Pritam Kumarsingh*: A Case Study to develop and test GIS/SDSS methods to assess the production capacity of a Cocoa Site in Trinidad and Tobago (2021).

133. *Muhammad Imran Khan*: Property Tax Mapping and Assessment using GIS (2021).

134. *Domna Kanari*: Mining geosocial data from Flickr to explore tourism patterns: The case study of Athens (2021).

135. *Mona Tykesson Klubien*: Livestock-MRSA in Danish pig farms (2021).

136. *Ove Njøten*: Comparing radar satellites. Use of Sentinel-1 leads to an increase in oil spill alerts in Norwegian waters (2021).

137. *Panagiotis Patrinos*: Change of heating fuel consumption patterns produced by the economic crisis in Greece (2021).

138. *Lukasz Langowski*: Assessing the suitability of using Sentinel-1A SAR multi-temporal imagery to detect fallow periods between rice crops (2021).

139. *Jonas Tillman*: Perception accuracy and user acceptance of legend designs for opacity data mapping in GIS (2022).

140. *Gabriela Olekszyk*: ALS (Airborne LIDAR) accuracy: Can potential low data quality of ground points be modelled/detected? Case study of 2016 LIDAR capture over Auckland, New Zealand (2022).

141. *Luke Aspland*: Weights of Evidence Predictive Modelling in Archaeology (2022).

142. *Luís Fareleira Gomes*: The influence of climate, population density, tree species and land cover on fire pattern in mainland Portugal (2022).

143. *Andreas Eriksson*: Mapping Fire Salamander (Salamandra salamandra) Habitat Suitability in Baden-Württemberg with Multi-Temporal Sentinel-1 and Sentinel-2 Imagery (2022).

144. *Lisbet Hougaard Baklid*: Geographical expansion rate of a brown bear population in Fennoscandia and the factors explaining the directional variations (2022).

145. *Victoria Persson*: Mussels in deep water with climate change: Spatial distribution of mussel (Mytilus galloprovincialis) growth offshore in the French Mediterranean with respect to climate change scenario RCP 8.5 Long Term and Integrated Multi-Trophic Aquaculture (IMTA) using Dynamic Energy Budget (DEB) modelling (2022).

146. *Benjamin Bernard Fabien Gérard Borgeais*: Implementing a multi-criteria GIS analysis and predictive modelling to locate Upper Palaeolithic decorated caves in the Périgord noir, France (2022).

147. *Bernat Dorado-Guerrero*: Assessing the impact of post-fire restoration interventions using spectral vegetation indices: A case study in El Bruc, Spain (2022).

148. *Ignatius Gabriel Aloysius Maria Perera*: The Influence of Natural Radon Occurrence on the Severity of the COVID-19 Pandemic in Germany: A Spatial Analysis (2022).

149. *Mark Overton*: An Analysis of Spatially-enabled Mobile Decision Support Systems in a Collaborative Decision-Making Environment (2022).

150. *Viggo Lunde*: Analysing methods for visualizing time-series datasets in open-source web mapping (2022).

151. *Johan Viscarra Hansson*: Distribution Analysis of Impatiens glandulifera in Kronoberg County and a Pest Risk Map for Alvesta Municipality (2022).

152. *Vincenzo Poppiti*: GIS and Tourism: Developing strategies for new touristic flows after the Covid-19 pandemic (2022).

153. *Henrik Hagelin*: Wildfire growth modelling in Sweden - A suitability assessment of available data (2023).

154. *Gabriel Romeo Ferriols Pavico*: Where there is road, there is fire (influence): An exploratory study on the influence of roads in the spatial patterns of Swedish wildfires of 2018 (2023).

155. *Colin Robert Potter*: Using a GIS to enable an economic, land use and energy output comparison between small wind powered turbines and large-scale wind farms: the case of Oslo, Norway (2023).

156. *Krystyna Muszel*: Impact of Sea Surface Temperature and Salinity on Phytoplankton blooms phenology in the North Sea (2023).

157. *Tobias Rydlinge*: Urban tree canopy mapping - an open source deep learning approach (2023).

158. *Albert Wellendorf*: Multi-scale Bark Beetle Predictions Using Machine Learning (2023).

159. *Manolis Papadakis*: Use of Satellite Remote Sensing for Detecting Archaeological Features: An Example from Ancient Corinth, Greece (2023).

160. *Konstantinos Sourlamtas*: Developing a Geographical Information System for a water and sewer network, for monitoring, identification and leak repair - Case study: Municipal Water Company of Naoussa, Greece (2023).

161. *Xiaoming Wang*: Identification of restoration hotspots in landscape-scale green infrastructure planning based on model-predicted connectivity forest (2023).

162. *Sarah Sienaert*: Usability of Sentinel-1 C-band VV and VH SAR data for the detection of flooded oil palm (2023).

163. *Katarina Ekeroot*: Uncovering the spatial relationships between Covid-19 vaccine coverage and local politics in Sweden (2023).

164. *Nikolaos Kouskoulis*: Exploring patterns in risk factors for bark beetle attack during outbreaks triggered by drought stress with harvester data on attacked trees: A case study in Southeastern Sweden (2023).

165. *Jonas Almén*: Geographic polarization and clustering of partisan voting: A local-level analysis of Stockholm Municipality (2023).

166. *Sara Sharon Jones*: Tree species impact on Forest Fire Spread Susceptibility in Sweden (2023).

167. *Takura Matswetu*: Towards a Geographic Information Systems and Data-Driven Integration Management. Studying holistic integration through spatial accessibility of services in Tampere, Finland. (2023).

168. *Duncan Jones*: Investigating the influence of the tidal regime on harbour porpoise *Phocoena phocoena* distribution in Mount's Bay, Cornwall (2023).

169. *Jason Craig Joubert*: A comparison of remote sensed semi-arid grassland vegetation anomalies detected using MODIS and Sentinel-3, with anomalies in ground-based eddy covariance flux measurements (2023).

170. *Anastasia Sarelli*: Land cover classification using machine-learning techniques applied to fused multi-modal satellite imagery and time series data (2024).