Masters Program in Geospatial Technologies

ENHANCING TEMPORAL SERIES OF SENTINEL-2 AND SENTINEL-3 DATA PRODUCTS: FROM CLASSICAL REGRESSION TO DEEP LEARNING APPROACH

Anu Bhalu Shrestha

Dissertation submitted in partial fulfilment of the requirements
for the Degree of *Master of Science in Geospatial Technologies*

NOVA IMS Information Management School
Universitat Jaume I
Westfälische Wilhelms-Universität Münster

ENHANCING TEMPORAL SERIES OF SENTINEL-2 AND SENTINEL-3 DATA PRODUCTS: FROM CLASSICAL REGRESSION TO DEEP LEARNING APPROACH

Dissertation supervised by:
Filiberto Pla Bañón, PhD
Professor, Institute of New Imaging Technologies (INIT),
Universitat Jaume I (UJI),
Castellon de la Plana, Spain

Co-supervised by:
Rubén Fernández Beltrán, PhD
Institute of New Imaging Technologies (INIT),
Universitat Jaume I (UJI),
Castellon de la Plana, Spain

Co-supervised by:
Mário Silvio Rochinha de Andrade Caetano, PhD
Associate Professor, Nova Information Management School,
Universidade Nova de Lisboa (UNL),
Lisbon, Portugal

February 21, 2021

ACKNOWLEDGEMENTS

I would like to extend my deepest appreciation to my supervisor Prof. Dr. Filiberto Pla Bañón for his continuous guidance and supervision since the inception of this thesis and for taking care of my overall well-being along with my work. I owe equal gratitude to my co-supervisor, Dr. Rubén Fernández Beltrán for his invaluable and generous support in all aspects of the thesis work and for creating a very encouraging working environment. I have received so much knowledge and positive energy, working with them was truly an honour. Similarly, I am grateful to my co-supervisor, Prof. Dr. Mário Silvio Rochinha de Andrade Caetano for his insightful suggestions and feedback on my thesis.

I am thankful to Damián for his assistance in solving the various problems faced during the thesis period. I cannot remain without thanking my friends and flatmates, Ganesh mama, Poshan dai and Janak, for letting me be the kid of the house and keeping me well-fed and well-motivated throughout this thesis. Thank you, my friend and my brother, Saadoon, for your inspiring words and also for the annoying but funny words that equally got me through difficult times in this new normal. I convey my gratefulness to my UJI geotech family for all the beautiful memories we created together in this masters course, and I express my due regards to Erasmus Mundus Program for this opportunity.

And to my family, my mom and my sisters, I am forever indebted, they have always been my support system showering me with their unparalleled love. I am specially grateful to David for his continuous encouragement and for persistently being there by my side at times of need. Thank you my friends, Samiksha and Pragya for cheering me up and motivating me and my spring day, Heather, for always bringing out the best in me. I dedicate this thesis to my family and Beyond The Scene.

ENHANCING TEMPORAL SERIES OF SENTINEL-2 AND SENTINEL-3 DATA PRODUCTS: FROM CLASSICAL REGRESSION TO DEEP LEARNING APPROACH

ABSTRACT

The free and open availability of satellite images covering global extent in recent days provides many novel opportunities for global monitoring of the earth's surface. Sentinel-2 (S2) and Sentinel-3 (S3) satellite missions capture mid to high resolution imagery with frequent revisit and show data synergy as they both focus on land and ocean observational needs. Specifically, the high temporal resolution of S3 (1-2 day revisit) presents potential in filling the data gaps in S2 (5 day revisit) vegetation products. In this scenario, this study assesses the feasibility of using Sentinel-3 images for Sentinel-2 vegetation products estimation using machine learning (ML) and deep learning (DL) approaches. This study employs four state of the art ML regression algorithms, linear regression, ridge regression, Support Vector Regression (SVR) and Random Forest Regression (RFR) and two DL network architectures with different depth and complexities, Multi-Layer Perceptron (MLP) and Convolutional Neural Network (CNN) to predict the S2 NDVI and SAVI maps from the S3 spectral bands information. A paired S2/S3 dataset is prepared for the study area covering one S2 tile in Extremadura, Spain. The results demonstrate that all the DL architectures except pixel-wise MLP outperformed the ML models with the 3D CNN performing the best. The best performing 3D CNN architecture obtained remarkable mean squared error (MSE) of 0.00198 for NDVI and 0.00282 for SAVI while the best performing ML algorithms were patch-wise RFR with MSE of 0.0035 in case of NDVI and patch-wise SVR with MSE of 0.00586 for SAVI. The models and the dataset prepared for this study will be useful for further research that focus on capitalizing the free and open availability of Sentinel-2 and Sentinel-3 imagery as well as new and advanced technologies to provide better vegetation monitoring capabilities for our planet.

KEYWORDS

Sentinel-2

Sentinel-3

Inter-sensor Data Products Estimation

Machine Learning

Deep Learning

Convolutional Neural Network

ACRONYMS

Adam Adaptive Moment Estimation.

BOA Bottom-of-Atmosphere.

CNN Convolutional Neural Network.

DL Deep Learning.

EC European Commission.
ESA European Space Agency.
EU European Union.

FC Fully Connected.

L1B Level-1B.
L2A Level-2A.
LTA Long Term Archive.

MAE Mean Absolute Error.
ML Machine Learning.
MSE Mean Squared Error.
MSI MultiSpectral Instrument.

NDVI Normalized Difference Vegetation Index.
NIR Near Infrared.

OLCI Ocean and Land Colour Imager.

RBF Radial Basis Function.
ReLU Rectified Linear Unit.
RMSE Root Mean Squared Error.

S2 Sentinel-2.
S3 Sentinel-3.

SAVI Soil Adjusted Vegetation Index.
SGD Stochastic Gradient Descent.
SNAP Sentinel Application Platform.
SVM Support Vector Machine.
SVR Support Vector Regression.
SWIR Shortwave Infrared.

TOA Top-of-Atmosphere.

VI Vegetation Index.
VNIR Visible and Near Infrared.

INDEX OF THE TEXT

INDEX OF TABLES

INDEX OF FIGURES

1 INTRODUCTION

1.1 Contextual Background

Sentinel-2 (S2) and Sentinel-3 (S3) satellite missions have become a source of unprecedented remote sensing data stream for the global monitoring of the earth's surface [1]. These two missions along with other satellites of Sentinel family make up the space component of Copernicus, the earth observation programme developed by the European Union (EU), with the involvement of the European Commission (EC), the European Space Agency (ESA), the EU Member States and Agencies. With the tagline of "Europe's eyes on Earth", it mobilizes its resources to capture and provide information on our planet for global environmental and security monitoring leading to a sustainable future [2]. It is also contributing in fulfillment and monitoring of the Sustainable Development Goals (SDGs) set by the United Nations by providing full, free and open access to the data collected by the Sentinels [3]. Specifically, S2 and S3 missions are focused on land and ocean observational needs imparting ways to retrieve and monitor various biophysical parameters related to cropland, forest and aquatic ecosystems that can help with the SDGs of zero hunger [4], life on land [5] and life below water [6].

S2 is a constellation of two identical sun-synchronous satellites, phased at 180° to each other and both carrying the MultiSpectral Instrument (MSI) sensor. Among the two satellites, S2A was launched on June 23, 2015 and S2B on March 7, 2017. The on-board MSI is a high resolution optical sensor with a versatile range of 13 bands ranging from the Visible and Near Infrared (VNIR) to the Shortwave Infrared (SWIR). These bands are acquired at varying spatial resolutions: four bands (B2-B4 and B8) at 10 m, six bands (B5-B7, B8A, B11 and B12) at 20 m and three bands (B1, B9 and B10) at 60 m. With these spectral and spatial features and a high revisit of 5 days, S2 aims to provide enhanced continuity to the multispectral imagery captured by SPOT and Landsat with improved data availability for users [7]. Another key mission objectives of S2 is to produce global land cover and land change detection maps and geophysical parameters which will be used for land management by European and national institutes, and the agricultural industry and forestry. Similarly, S3 mission consists of a pair of identical satellites: S3A, launched on February 16, 2016 and S3B, which followed on April 25, 2018. Among the different instruments carried by S3, Ocean and Land Colour Imager (OLCI) is one of the optical sensors, that captures 21 spectral bands (Oa01–Oa21) in the VNIR range of 390 to 1040 nm with a spatial resolution of 300 m across all bands. The two S3 satellites working together enable a short revisit time of less than two days for OLCI. Its objective is the global

measurement of the sea and land topography, temperature, and color with a particularly high temporal resolution giving continuity to ENVISAT and SPOT-Vegetation products [8] for environmental, climate and vegetation monitoring. Thus, these two missions show a data synergy in global land monitoring through the use of mid to high resolution multispectral optical energy [9].

Currently, multi-sensor and inter-sensor platforms are becoming important to utilize available satellite data resources for innovations beyond their primary objective and to fulfill the demand of applications that require as temporally dense data as possible [10]. For instance, many regional and global applications such as Mars Crop Yield Forecasting Systems (MCYFS) of EC and Group on Earth Observations (GEO) Global Agricultural Monitoring (GEOGLAM) crop monitor, use time series of the satellite-derived vegetation product, Normalized Difference Vegetation Index (NDVI), to monitor crop condition and yield estimation. In this scenario, the synergy between S2 and S3 gives an exceptional opportunity for the scientific community to develop new models and methodologies to deal with the problems of data gaps and enhance the temporal series of data products from each mission [11][12]. In particular, the high temporal resolution of S3 shows potential in the improvement of S2 NDVI time series and similar vegetation products, that contributes in increasing the quality of vegetation monitoring applications. Similarly, the high spatial resolution of S2 can help to enrich the spatial resolution of S3.

Estimating inter-sensor data or data products is a regression problem that can be solved by fitting a model that maps the relation between the input (eg. reflectances) and the output (eg. reflectances or parameters). Such inter-sensor data estimation is specially useful in case of enhancing the temporal resolution of imagery or data product from one sensor by filling the data gaps caused by lower temporal resolution of the sensor or external factors such as cloud coverage and atmospheric conditions by the use of data from another sensor. Recently, Machine Learning (ML) and Deep Learning (DL) techniques have become an important tool in many fields including remote sensing due to its high success rate in image classification, object detection and semantic segmentation [13]. Such models have also performed well in regression tasks such as estimating biophysical parameters [14], vegetation indices [15] and crop yield prediction[16]. Thus, it is important to explore and utilize the potential of machine and deep learning technology in developing models for inter-sensor data products estimation that will enhance the capabilities of the sensors.

1.2 Problem Statement and Motivation

Considering the suitability of the sensors, many approaches have been applied to estimate the vegetation products such as biophysical parameters and vegetation indices in the context of Sentinel missions. However, most of the past studies performed have used linear and ML regression approaches and often they are conducted from a single sensor perspective and using simulated data [17]. In addition to that, even though other techniques such as constrained topic modelling [11] has been used, DL technology has not been implemented in inter-sensor data products estimation in case of S2 and S3. The

study done by [12] to predict S3 data from S2 shows that non-linear regression approaches perform better than linear methods in modeling the relationship between S3 and S2 data. Similarly, there have been multi-sensor related works combining S2 with Sentinel-1 or other missions such as Landsat [15] [18] but this has not been explored for S2 and S3 using deep learning.

The free and open availability of S2 and S3 data, their data synergy, high temporal resolution of S3, and the availability of novel DL architectures and advancements in remote sensing motivates this study to explore the possibility of inter-sensor data product prediction models. The successful application of this study in case of S3-S2 inter-sensor vegetation product estimation will encourage more extensive research extending to the prediction of more complex products such as biophysical products as well as other sensors of Sentinel or other missions. This will contribute in enhancing the vegetation monitoring capabilities of the satellite sensors.

1.3 Research Aims and Objectives

The main aim of this research is to enhance the temporal series of Sentinel-2 vegetation products by using the overlapping series of Sentinel-2 and Sentinel-3 imagery with the help of classical regression algorithms and deep learning models. The following specific objectives have been set to fulfill this aim:

- To select suitable deep learning architecture by reviewing existing deep learning algorithms for inter-sensor vegetation products estimation.
- To implement the selected models and optimize the performance of the deep learning architectures.
- To compare the performance of the selected deep learning models with state-of-art classical regression algorithms.

1.4 General Methodology

The overall methodology of this thesis is composed of four stages that are: i) review and choice of network architecture, ii) dataset download and preparation, iii) design, and implementation of prediction models, and iv) performance comparison. The schematic diagram at 1.1 shows the detailed structure of the methodology.

Firstly, several existing ML and DL approaches that can be used for vegetation product prediction from Sentinel and other satellite imagery were reviewed. State of the art DL network architectures with different depth and complexities, Multi-Layer Perceptron (MLP) and Convolutional Neural Network (CNN) are selected. Along with this, four classical machine learning algorithms used for similar regression problems, linear regression, ridge regression, Support Vector Regression (SVR) and Random Forest Regression (RFR) are chosen.

In the second stage, the dataset to be used for training the DL and ML models was prepared. The study area is a S2 tile in Cáceres and Badajoz province, Extremadura region of Spain. S2 Level-2A (L2A) products and S3 OLCI Level-1B full resolution products are downloaded and further processed to prepare the images for training the models.

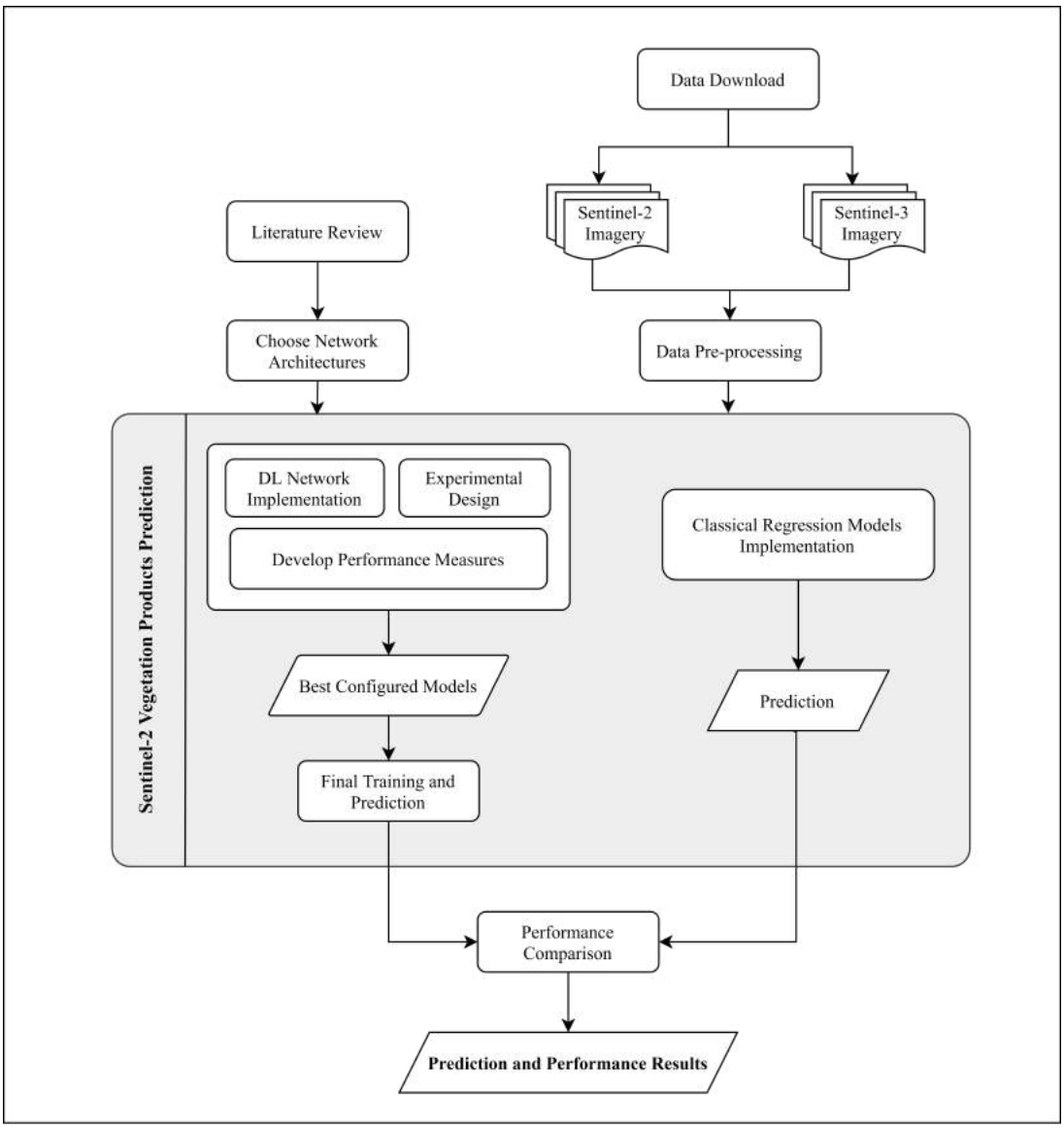


Figure 1.1: Methodological overview

The third stage deals with the design, implementation, and analysis of the chosen Convolutional Neural Network (CNN) architectures. Multi-layer perceptron and CNN architectures were implemented to inspect how the performance of the deep learning algorithms vary with the variations and increasing depth in the network architecture. Experiments were performed to select the optimal values of the hyperparameters. Similarly, all four ML algorithms were also implemented.

Finally, in the last phase, the performance of all the models was evaluated using quantitative and qualitative approaches. For a quantitative approach, three metrics namely Root Mean Squared Error (RMSE), Mean Squared Error (MSE) and Mean Absolute Error

(MAE) were used as measures of evaluation. Additionally, visual inspection of predicted vegetation product maps was used as a means for qualitative evaluation.

1.5 Contribution

This study assesses the feasibility of using S3 imagery to fill the data gaps in S2 vegetation products using an inter-sensor approach. The main contribution of this thesis consists of:

- Exploring the possibility of using Sentinel-3 data to effectively recover Sentinel-2 data products.
- Validating the suitability of various regression and deep learning approaches in successfully predicting S2 vegetation products from S3 data.
- Comparison of chosen network architectures and finding the best performing model.
- Preparation of a paired S2/S3 data set of the study area for the year 2019 that can be used for further research purposes as there is lack of such inter-sensor collections in the RS community.

1.6 Thesis Organization

This thesis consists of seven chapters. Chapter 1 introduces the contextual background of the thesis with the problem statement and motivation behind the work, aims and objectives and highlights the contribution of the work. Chapter 2 reviews the related works on the existing methods for regression modelling for vegetation products estimation using remote sensing imagery. Also, this chapter deals with the selected algorithms to be implemented to fulfill the objective of the study. Chapter 3 provides the theoretical background of the chosen DL network architecture along with the ML algorithms. Chapter 4 gives an overview of the dataset used as input for the models along with an explanation of the pre-processing steps applied and the resources used. Chapter 5 presents the methodological description of implementation, training, experimental settings and performance evaluation of chosen algorithms for regression purposes. In Chapter 6, the results of the experiments are shown, interpreted and discussed. Finally, Chapter 7 deals with the conclusion drawn from this thesis and recommendations for future work directions.

2 LITERATURE REVIEW

This chapter provides a review on the vegetation products estimation from satellite imagery with a focus on Sentinel-2 (S2) and Sentinel-3 (S3) sensors with related works based on machine learning and deep learning techniques in the first and second sections. In the last section, the choice of deep learning architectures and machine learning algorithms to be used in this study is explained.

2.1 Vegetation Products Estimation

Satellite sensors remotely capture the electromagnetic radiation reflected by the vegetation containing information on their biophysical composition that can help in deriving various vegetation products [19]. These vegetation products include biophysical parameters such as leaf area index (LAI), fraction of absorbed photosynthetically active radiation (FAPAR), leaf chlorophyll content as well as vegetation indices (VI) such as Normalised Difference Vegetation Index (NDVI) and Soil Adjusted Vegetation Index (SAVI). The biophysical parameters are essential inputs to crop and forest monitoring systems as they provide valuable insight regarding vegetation growth, health status and productivity [20][21]. The VIs, on the other hand, are computationally simple and effective measures, designed based on vegetation reflectance properties to accentuate the spectral properties of the green plants [22]. VIs such as NDVI has been used for vegetation and landcover change detection [23] and SAVI for crop yield estimation [24]. VIs are also one of the commonly used methods to retrieve the biophysical products on a regional or global scale [25].

S2 MSI and S3 OLCI sensors have been designed with multiple spectral bands around the chlorophyll absorption and red edge region making them highly suitable for vegetation products estimation purpose. The study by [1] highlights the potential of S2 and S3 in meeting the various satellite observational needs including the capabilities to derive the FAPAR, LAI, and NDVI for land observations. To benefit from the enhanced sensor configurations of S2 and S3 compared to their predecessors, Landsat, SPOT and MERIS, researches have been performed for improved vegetation parameter retrieval methods. Research based on experimental S2 and S3 data simulated from hyperspectral imagery to estimate leaf chlorophyll content, LAI and fractional vegetation cover (FVC) showed that these parameters can be accurately mapped from S2 and S3 spectral bands [17]. A comparison of NDVI products of S2 and MODIS with ground based sensors demonstrated that S2 based NDVI is better correlated with ground measurements [26]. Similarly, S2 derived NDVI and SAVI are also found to be correlated with and an efficient method for

crop yield prediction [27][28]. Some studies feature the use of both S2 and S3 either to perform a data fusion for higher resolution image and data products [29] or to compare their data products. The study to estimate Chlorophyll-a over inland water presented good accordance between the S2 and S3 derived Chlorophyll-a maps in terms of magnitude and spatial distribution [30]. These researches exemplify the appropriateness of S2 and S3 in vegetation products estimation and the data synergy between them. Thus, S2 and S3 data can be used for inter-sensor data or data products estimation of each other to utilize the specific benefits of each sensor in spatial, spectral and temporal resolution.

2.2 Related works

2.2.1 Classical Machine Learning Approach

Machine learning techniques provide a collection of various linear and non-linear regression methods that have been used in conjunction with satellite data for purposes ranging from biophysical parameter retrieval [17], landcover classification [31] to crop yield estimation [32]. Among the various methods used for biophysical parameter estimation, ML approaches are non-parametric and they utilize the spectral information from all available bands in mapping the relationship from input reflectance to output without any assumptions [33]. A study done by [33] presented a comparison among parametric, ML and physical retrieval methods for S2 LAI estimation and concluded that the ML approaches are best suited for accurate and fast retrieval of vegetation properties. Kernel based methods, Gaussian Process Regression (GPR), Support Vector Regression (SVR) and Kernel Ridge Regression (KRR) generated robust results for biophysical parameter retrieval using simulated S2 and S3 data[17] [34]. While the study by [35] showed that linear regression and Random Forest Regression (RFR) provided the best results in estimating LAI and leaf chlorophyll content respectively from S2 satellite data. Even though there are many research articles in this field, very few of the research is focused on the regression problem from inter-sensor perspective, that is estimating the data or data products of one sensor given the data of another sensor. One of such related works is based on constrained topic modelling by [11] to estimate S3 vegetation indices NDVI and SAVI from S2 image data. Another study was done by [12] to restore S3 image from S2 data which shows that ridge regression and a non-linear extreme learning machine performed better than principal component analysis method in modeling the relationship between S3 and S2 data and generated accurate results.

2.2.2 Deep Learning Approach

Deep learning architectures, specifically the Convolutional Neural Networks (CNN) have outperformed the existing state of the art machine learning methods for computer vision tasks [36]. Even though CNNs have been mostly applied for image classification and object detection, they are applicable for deep learning regression tasks as well [37]. The ability of CNNs to effectively extract the important features, to represent spatial patterns and its

flexible architecture design makes it the DL architecture of choice in various classification and regression tasks in remote sensing [38] [39]. As shown by the review of literature, application of deep learning techniques in this field is quite new and dominated by the studies on satellite image classification, object detection and segmentation. With the focus on vegetation remote sensing, a recent review on use of CNN, [39] reported that more than 75 percent of the related studies were published in 2019 or later and only 10 percent studied image regression approach, the most common approaches being segmentation and classification. Similar to ML, deep learning regression has been applied in estimating biophysical parameters [40][14], vegetation indices [15] and crop yield prediction [16]. [40] employs a mixed approach by combining visual features of smartphone captured images via convolutional branch and the ground based plant information via Multilayer Perceptron (MLP) branch for LAI estimation in wheat and the estimations had high correlation with ground values. Another study in biophysical parameter estimation predicted global chlorophyll-a concentration from MODIS imagery using four layer patch based CNN and concluded that it outperformed the classical SVR [14]. The authors of [15] proposed a data fusion approach based on CNN to estimate the NDVI from various combinations of multi-temporal Sentinel-1 SAR and S2 images. The results highlighted the capability of the proposed three layer patch based CNN architecture to capture the relationship between SAR and NDVI and focused on the proposed framework being a general one that can be easily extended for estimating other spectral features. Satellite data is characterized by its temporal resolution in addition to the spectral and spatial resolutions. In CNNs, it is the most common approach to apply filters to the spatial dimensions to generate two dimensional feature maps [13]. Three dimensional (3D) CNN [41] architectures help to capture the temporal dimension of multi-temporal image data for spatio-temporal feature learning and such architectures have been used in crop classification [42] and crop yield estimation [16] with accuracy outperforming the existing state of the art.

2.3 Selection of Models

From the literature review, it is seen that ML algorithms have been commonly used to solve vegetation remote sensing related regression problems, however DL application in this field is quite new. For the purpose of this study, four of the most frequently used standard ML regression algorithms, linear regression, ridge regression, SVR and RFR and two DL network architectures MLP and CNN are chosen for the inter-sensor vegetation products prediction. These architectures are chosen to analyse the effect of network structure and complexity on the performance of the models since they have different depth and complexities. A simple MLP architecture with a single hidden layer is adapted from the study by [43] where it has been used for classification task using hyperspectral imagery. Within CNN, four types of networks are considered. The first one is a basic 2D CNN architecture based on the classical LeNet-5 with two convolution and pooling layers and two fully connected (FC) layers [44] and is similar to the structures used in [14] and [15]. The second CNN is chosen for its deeper architecture with five

convolution and two FC layers and adapted from [45]. Most of DL architectures and studies available often focus on high resolution imagery [46] and S3 imagery used in this study is mid resolution. So the structure by [45] is chosen as this was designed specifically for mid resolution sensors. Finally, the last two architectures implemented are 3D versions of the 2D CNNs. They are studied to create 3D temporal models utilizing multi-temporal data as input and for a comparison with their 2D counterparts.

3 THEORETICAL BACKGROUND

This chapter presents the background concepts used in this research. The first section provides a brief explanation about the vegetation products chosen for prediction in this study. Next, classical machine learning regression approaches are discussed. The last section describes artificial neural networks and deep learning concepts in detail.

3.1 Vegetation Products (NDVI and SAVI)

Normalised Difference Vegetation Index (NDVI) and Soil Adjusted Vegetation Index (SAVI) are the vegetation products chosen for this study, however the methodology can be extended to other vegetation indices or other products. NDVI and SAVI are Vegetation Index (VI) used as simple and effective measures to provide insight on vegetation cover, growth and health from remotely sensed information [22]. They are designed based on vegetation reflectance properties to accentuate the spectral properties of the green plants. Two (or more) different bands, often in the plant absorption and reflection spectrum, that is visible RGB from approximately 400-700 nm and NIR wavelengths from approximately 700-1300 nm respectively, are combined to form such indices[19].

Normalised Difference Vegetation Index (NDVI) is one of the most widely used VI for the purposes ranging from biophysical parameter estimation to landcover classification and crop health monitoring [47]. The NDVI is a numerical indicator which uses visible and near-infrared as there is high interaction of energy in the visible and near infrared regions of the electromagnetic spectrum. Generally, healthy vegetation absorbs most of the visible light and reflects a large portion of near-infrared light while unhealthy or sparse vegetation reflects more visible light and less near-infrared light. Bare soils on the other hand reflect moderately in both portion of the electromagnetic spectrum. The mathematical equation to calculate NDVI is given as:

$$NDVI = \frac{(\rho_{nir} - \rho_{red})}{(\rho_{nir} + \rho_{red})} \tag{3.1}$$

where ρ_{nir} represents the reflectance of the near-infrared band and ρ_{red} represents the reflectance of the red band. Since the value of reflectance ranges from 0 to 1, the value range of NDVI is -1 to 1. Values close to zero (0.1 or less) generally correspond to areas barren rock, sand, or snow, sparse vegetation such as shrubs and grasslands or senescing crops may result in moderate NDVI values (approximately 0.2 to 0.5) while high NDVI

values (values approaching 1) indicate dense vegetation or crops at their peak growth stage.

Soil Adjusted Vegetation Index (SAVI) is an improvement in the previous index, NDVI to account for soil brightness [48]. It is specially useful in the areas where the vegetation cover is low. It was used alongside NDVI for vegetation delineation and yield estimation [49] [24]. SAVI is calculated as:

$$SAVI = \frac{(\rho_{nir} - \rho_{red})}{(\rho_{nir} + \rho_{red} + L)} * (1 + L) \tag{3.2}$$

where, L is the soil brightness correction factor that ranges from 0 to 1. The value 0 indicates very high vegetation cover while the value 1 indicates very low vegetation cover.Generally, the value of 0.5 is mostly used representing the intermediate vegetation cover.

To calculate these indices using Sentinel-2 data, band number 4 (B4) is taken as red band and band number 7 (B7) is NIR band and the brightness correction factor is set to 0.428 [50] [51].

3.2 Machine Learning Regression Algorithms

Machine learning (ML), a subset of artificial intelligence, gives systems the ability to learn and optimize processes without being specifically programmed for the task. It is an effective empirical approach applied widely for both regression and/or classification problems [52]. Regression analysis is a basic concept under supervised machine learning where the algorithm is trained with both input features and output labels/values to establish the relationship between the dependent and independent variables. These models map the input space to a real value domain. In supervised ML, an iterative procedure is used to minimize the prediction error of a ML model for the given training data. The error, called loss function, is usually expressed as a difference between output predicted by the model and actual output or label given as a part of training data by applying optimization methods. Different machine learning algorithms can be implemented using different models, loss functions and optimization methods.

3.2.1 Linear Regression

Linear Regression is an algorithm for supervised machine learning where the forecast output is continuous and has a steady slope. The relationship between two or more variables is modelled by a linear equation in the dataset. Independent variables are the features that are provided as input data and dependent variables are the target that are to be predicted. This regression problem is represented as:

$$y = \sum_{j=1}^{n} w_j x_j + b \tag{3.3}$$

where, y is predicted value, b the bias, w_i is the coefficient or weight of i^{th} feature and x_i is the input value for i^{th} feature and n is the total number of input features. MSE and MAE are commonly used loss functions for linear regression.

3.2.2 Ridge Regression

Ridge Regression is a linear regression extension that applies a regularization penalty to the loss function during training depending on the sum of the square coefficient values to prevent overfitting [53]. When the model memorizes the training data rather than finding patterns, it will not generalize well on new data which is termed as overfitting. Regularization is a technique used to control the this phenomenon. It works by penalizing the magnitude of feature coefficients and minimizing the error between the projected observations and the real observations. Coefficients evidently increase to fit with a complex model, so when penalized, it puts a check on them to avoid such scenarios. For those input variables that do not contribute much to the prediction task, this has the effect of shrinking the coefficients.

3.2.3 Support Vector Regression

Support Vector Regression (SVR) is a regression technique based on Vapnik's concept of support vectors [54][55]. It is an extension of Support Vector Machine (SVM) algorithm which was introduced for regression scenarios. While SVM generates a hyperplane for class label prediction, SVR predicts numerical values using a function derived from the training data. Similar to SVM, SVR also projects training data into higher-dimensional space representations in a given feature space in which a linear regression function can be derived. This is accomplished through the use of kernel functions such as Radial Basis Function (RBF). Using Vapnik's ε-insensitive approach, only the errors beyond ε are penalized. SVR's computational complexity is independent of the dimensionality of the input space.

3.2.4 Random Forest Regression

Random forest is an ensemble learning technique for classification or regression that combines multiple decision trees into a forest or final model to produce more accurate and stable outputs [56]. In case of classification, the output is mode of the classes and for regression, the average prediction of the individual trees. The theory behind the techniques of ensemble learning is based on the assumption that its accuracy is greater than other machine learning algorithms because the combination of predictions works more reliably than any single constituent model.

The decision trees in RF regression are regression trees that represent a collection of conditions or constraints that are hierarchically ordered and applied successively from a root to a tree leaf. The RF starts with several randomly drawn bootstrap samples with replacements from the original training dataset. Each of the bootstrap samples is fitted with a regression tree. For each node per tree, a small set of input variables selected from

the total set is randomly considered for binary partitioning. The decision tree splitting criterion is based on choosing the attribute with the lowest value of an index measuring the impurity, such as Gini Impurity Index (IG) [57].

3.3 Deep Learning and Artificial Neural Networks

Deep learning (DL) is a subfield of ML based on an algorithm called Artificial Neural Network (ANN). ANNs are modeled after the way the human brain works, aiming to solve problems through extensive learning from data while DL deals with the approach and technique of learning in neural networks [58]. ANNs are composed of a high number of interconnected computational nodes, commonly referred to as neurons [59]. An ANN architecture generally consists of an input layer, an output layer, and one or more hidden layers between them, and the number of layers is known as the depth. Even though DL is also used to refer to such ANN architectures with many hidden layers, it is also about the entire architecture, processing functions, and the regularization techniques used to facilitate learning [60]. Various forms of DL architectures such as supervised - convolutional and recurrent neural networks, and unsupervised - autoencoders, have been implemented and achieved unprecedented accuracy in diverse fields ranging from computer vision, to natural language processing, and medical image analysis. A multilayer perceptron consisting of a basic ANN architecture and a deeper CNN architecture are the considered DL algorithms for this study.

3.3.1 Multilayer Perceptron

Multilayer perceptron (MLP) is the simplest kind of feed-forward artificial neural network consisting of at least three layers of units: an input layer, a hidden layer and an output layer [61]. Every unit in one layer is connected to every unit in the next layer such that the network is fully connected. Each unit in MLP,except for the input units, is a neuron that uses a nonlinear activation function.The neurons in the first layer lead to the problem's independent input variables to be dealt with and pass the input values to the next layer.There may be one or more hidden layers that obtain the weighted combination of input values from the previous layer and, depending on their activation function, generate an output. The weights are determined and adjusted, through an iterative and a back-propagation process for training, minimizing an error function. Finally, the last output layer gives one unit for each value the network outputs, that is, a single unit for regression or more for classification problem. Since MLPs are fully connected, the number of total parameters can grow to very high with the increase in layers and features. It takes inputs in the form of flattened vectors disregarding the spatial information in the process.

3.3.2 Convolutional Neural Network

Convolutional neural network (CNN) is a natural extension to MLP with few modifications which has gained state-of-art status in the field of computer vision in tasks ranging from image classification to object detection and semantic segmentation [36]. The variations in

the form of adding convolution and pooling layers and the sparse connectivity reduced parameter dimensions and enabled local connectivity [62]. In CNN, a neuron in a hidden layer is connected only to a sub-region called as receptive field of the input making the network sparsely connected resulting in fewer parameters and lesser computations. Parameter sharing is an important property of CNN, where the same weighted connections are shared by all the neurons belonging to a specific feature map and these neurons cover all the receptive fields used.Instead of a single vector, CNNs treat images as a multidimensional input and specifically consider the spatial contexts of image pixels. These are some of the features that together enable CNN to model complex relationships among input elements efficiently. CNNs are also used to solve regression problems in which case, the softmax layer for classification is generally replaced with a fully connected regression layer with linear or sigmoid activations [37].

3.3.2.1 Architecture of a CNN

A CNN typically consists of a set of alternatively stacked convolutional and pooling layers followed by one or more fully connected layers. A convolutional block is composed of convolution, activation, and pooling layers.The convolution and pooling layers act as feature extractors of the input images. Finally, the completely connected layer, after a series of convolutions and pooling gives the predicted value (for regression) or class score of each pixel (for classification) in a feed-forward manner via the network.

Convolutional layer

The convolution layer is the main building block of the CNN that has a number of filters (kernels) and convolves them on an input image for extracting features corresponding to each receptive field and weight kernels. Mathematically, a matrix multiplication is carried out between the number of filters and the input feature size resulting in a 2-dimensional feature map. The number of convolutional layers can vary from one to many based on the number of datasets, feature complexities and computational capacity available. Present as hidden layers, they are mainly responsible for extracting features such as edges, colors, orientation of the input data and reducing the image size to ease the learning process with no loss in data properties. The convolution operation generally downsamples the output image size by an amount that depends on the filter and stride size. The spatial resolution of the original input can be preserved by introducing zero-padding, a process of wrapping the input image with zeros.

Activation functions

Activation functions are the functions that introduce non-linearity into the network to enable nonlinear representations without affecting receptive fields of the previous convolutional layer. These are also considered as the decision function of the neuronś output. Activation layers are often placed right after a convolutional layer, but many combinations are possible. A variety of function are popular and being adopted for as activation function

such as the sigmoid function, Tanh, and Rectified Linear Unit (ReLU). Sigmoid and Tanh functions tend to saturate when initialized weights are too high or if gradient tends to zero, they are prone to the vanishing gradient problem. New non-saturating nonlinearities, such as ReLU [63] has been proposed to alleviate this. A ReLU activates by thresholding the negative inputs to zero and passing the positive inputs unchanged. ReLU is proven to be computationally efficient and effective for convergence.

Pooling layer

The pooling layer performs sub-sampling along the spatial dimensions of feature maps using predefined functions on a local region or receptive field to summarize the signal spatially preserving discriminant information. It provides a form of robustness to the network by improving translation invariance and reduces the computational cost of the network by discarding redundant information and reducing the spatial resolution of feature map. The standard pooling strategies are maximum pooling, known as Max-pooling, and average pooling. The former returns the maximum values in the receptive field while latter returns the average of the group of activation over the receptive field. Max-pooling is usually considered a better option than the average pooling.

Fully Connected Layer

In a Fully Connected (FC) layer, neurons have connections to all the activations in the preceding layers which is analogous MLP architectures. After several convolutions and pooling layers, FC layers are normally positioned at the end of the network. FC layer takes the output of the previous layers, flattens them into a vector that can be input to the next stage. For regression problems, the FC layer with a single node and a linear activation function acts as the regression layer. If there are multiple FC layers, the last FC layer gives the final predicted value.

Regularization

Regularization techniques are used to avoid overfitting and decrease the generalization error of the data in the network. Overfitting in deep learning networks where networks occurs when they fit too well to the training data but cannot generalize, resulting in large gap between the training and test errors. There are many methods for regularization such as L2 and L1 regularization, dropout, and early stopping. L1 regularization makes the model sparse and that only contributes to regularize the model to less extent and and is thus not always used. L2 regularization (also known as weight decay) is widely used regularization techniques that minimizes the sum of the square of the differences between the target values and the estimated values. Dropout works by randomly knocking out units in the network resulting in a smaller network. Early stopping is another powerful regularization technique when training the neural networks to stop the training of the neural network at a point where the performance on a validation dataset begins to degrade.

3.3.2.2 Training

CNN's training process consists of forward computation, loss optimization, and back-propagation and parameter updating.

Forward Computation

As described in the earlier section, the input is fed through the neural network architecture consisting of a series of convolution, pooling, and completely connected layers.The network predicts the labels or values based upon the network architecture (classification or regression).

Loss Optimization

The network output needs to be optimized by changing the values of parameters that are learned by the network, such as weights and bias. The optimization problem describes the uncertainty in deciding the optimum set of parameters that the loss function quantifies. In the case of regression, the mean square error is often employed as the loss function that computes the sum of squared deviations between the target and predicted values.

Back-propagation

After the loss function is described, to extract the parameters that reduce the loss, training of the convolutional network must be done.The network attempts to reduce this error by changing the weights of neurons in each iteration via the back-propagation process, . Several optimization algorithms are available to gradually update the weight and bias in search for the optimal solution such as Stochastic Gradient Descent (SGD) with momentum, Adaptive Gradient (AdaGrad), Root Mean Square Propagation (RMSProp), and Adaptive Moment Estimation (Adam) among which Adam has been used for this study.

Adaptive Moment Estimation (Adam), proposed by [64], is an optimizer based on mini-batch gradient descent that computes adaptive learning rates for each parameter. It is an update to the RMSProp optimizer where momentum from SGD is incorporated. As in RMSProp, Adam also uses an exponentially decaying average of previous squared gradients and uses an exponentially decaying average comparable to momentum of previous gradients, where such a moving average of previous gradients helps to speed up intertia learning.

3.3.2.3 Parameters and Hyperparameters

Parameters are the variables that are used in the model that can effect the performance of the model based on the values selected for these parameters.The main parameters of deep neural networks are weights and biases. A hyperparameter is a parameter whose value is used to control the learning process and should be set before the training process.They need to be tuned for each problem because they are often problem and data specific. The method of finding the combination of hyperparameter values for a model that performs

the best as calculated on a validation dataset is called hyperparameter tuning. It is the problem of choosing a set of optimal hyperparameters for a learning algorithm.

Hyperparameters can be divided into two types: hyperparameters that determine the network structure such as kernel size, stride, padding hidden layers, and activation function and those that determine the network training process such as learning rate, batch size, number of epochs, regularization techniques and so on. Some of the influential hyperparameters are discussed in the following section:

Learning Rate:

The learning rate signifies how quickly the gradient updates to the parameter follows the gradient direction.The model takes a lot of time to converge due to a small learning rate, while a large learning rate causes the model to diverge and losses may fluctuate indefinitely.Starting with a high learning rate and lowering it as the training goes on was a general technique used to select the learning rate.

Batch size:

The batch size controls the number of training samples to work through before the model's internal parameters are updated. It optimizes the training of a network by defining how many samples to read at a time and keep in memory.

Number of epochs:

This hyperparameter controls the number of complete forward and backward passes over the training samples. After each epoch, the weights are updated and thus show better performance. Using several epochs, however, can overfit the training phase, and this is when early stopping can be used, preventing the model from overfitting.

3.3.2.4 3D Convolutional Neural Networks

The basic structure and workings of 3D CNN is similar to 2D CNN, with an added dimension for kernels and the convolution and pooling layers [41]. In 2D CNN, a 2D kernel moves in two dimensions along the width and height of the image space resulting in a 2D image output whereas the use of 3D kernels in 3D CNNs gives the freedom to move the kernel in three dimensions on the temporal stack of input images. The third dimension, depth of the kernel helps to preserve the temporal information resulting in 3D volume output after convolution and pooling operations.

4 DATASETS AND RESOURCES USED

This chapter provides a description of the data used in this study in its first section followed by a detailed explanation on the pre-processing steps in the second section. The final section presents a brief overview on the resources, including hardware and software used for the experimental set up of this research.

4.1 Data Description

The research uses a paired dataset of 41 S2 and S3 imagery of same day in the year 2019 over the area covered by one S2 granule, also called tile with *tileid* '30STJ' and relative orbit number 137. Figure 4.1 shows an overview of the S2 and S3 tiles.

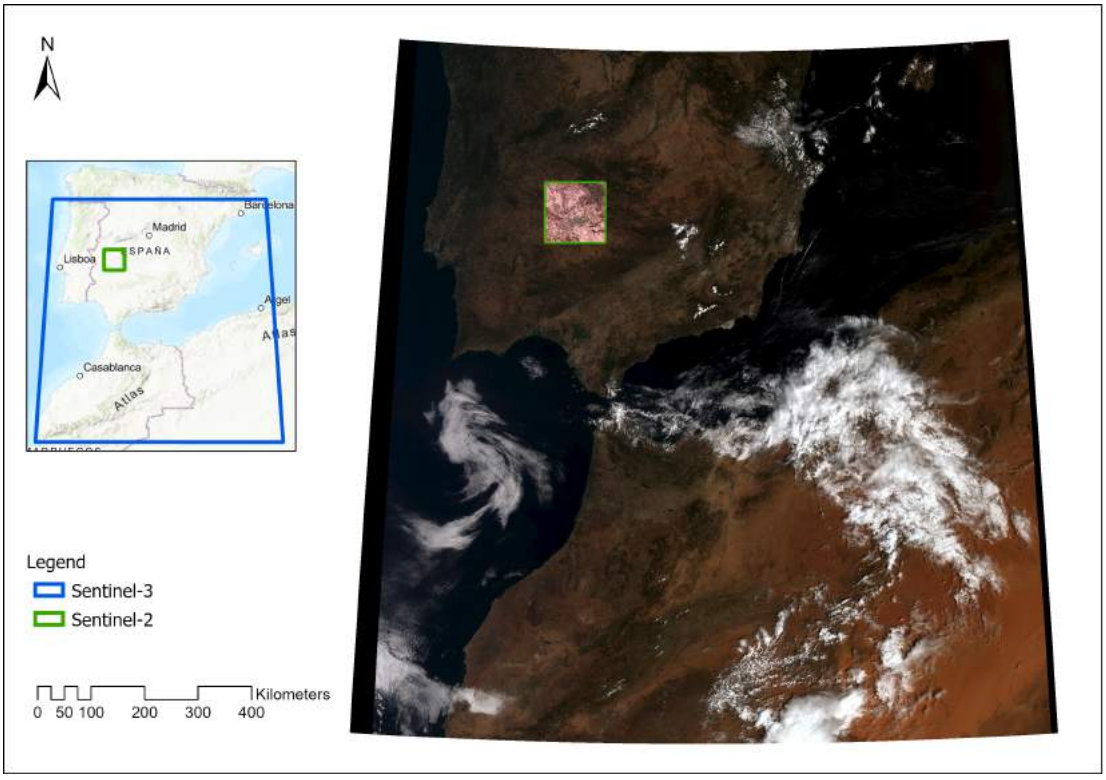


Figure 4.1: An overview of Sentinel-2 and Sentinel-3 tiles

This tile lies in Cáceres and Badajoz province, Extremadura region of Spain covering

100 x 100 km^2 area bounded by (-5.188652, 38.740370) and (-6.498977, 39.697509) latitude-longitude coordinates. S3 tiles cover a larger area compared to S2 as seen in figure 4.1), so they were clipped to fit S2 extent during pre-processing.

According to [65], Extremadura region has the highest percentage of forest area in Spain covering 65% of the total area and a low percentage (0.69%) of the artificial area. This area is important from agricultural point of view as well with the highly developed dehesa system [66], a form of agro-forestry made up of natural grasslands and oak forests. The study area land cover is also mostly agrro-forestry and natural grasslands, including some protected areas of Extremadura such as Parque Natural de Cornalvo and Zona de Interés Regional Llanos de Cáceres y Sierra de Fuentes.

4.1.1 Sentinel-2 Level-2A Product

Sentinel-2 products are available for the users in two levels: Level-1C (L1C) product with Top-of-Atmosphere (TOA) reflectance and Level-2A (L2A) products with Bottom-of-Atmosphere (BOA) reflectance [67]. This study uses L2A products that are atmospherically and radiometrically corrected. L2A products of Europe region are produced at the ground segment since March 2018, and globally since December 2018. These products can also be derived from the associated L1C products through the Sen2Cor processor developed by ESA. Each L2A product is composed of 100x100 km2 ortho-rectified image called tile or granule in projected UTM/WGS84 system. The studied tile lies in 30N UTM zone. The products are disseminated via Copernicus Open Access Hub [1] in Standard Archive Format for Europe Format (SAFE) file format [68]. S2 L2A product has 12 spectral bands with varying spatial resolution in the range from 10m to 60m (Table 4.1).

Table 4.1: S2 L2A spectral bands definition, Source: [67]

Band	Central wavelength (nm)	Bandwidth (nm)	Spatial Resolution (m)	Description
B1	443	20	60	Aerosols
B2	490	65	10	Blue
B3	560	35	10	Green
B4	665	30	10	Red
B5	705	15	20	Red Edge 1
B6	740	15	20	Red Edge 2
B7	783	20	20	Red Edge 3
B8	842	115	10	NIR
B8A	865	20	20	Red Edge 4
B9	945	20	60	Water Vapour
B11	1610	90	20	SWIR 1
B12	2190	180	20	SWIR 2

The four bands at 10 m spatial resolution: the Red, Green, Blue (RGB) and Near Infrared (NIR) bands are used for land applications. Among the six 20m bands, four red edge bands are for vegetation characterization and two SWIR bands are for detecting snow,

[1] https://scihub.copernicus.eu/

ice or cloud. The remaining two bands at 60m are for atmospheric corrections and cloud screening. The cirrus band (Band 10) is omitted from L2A product as it does not contain surface information.

The S2 images available over the study area for the year 2019 were filtered on the basis of cloud cover percentage. As a result 41 images with less than 20% cloud cover were obtained. The L2A product comes with a scene classification map with clouds and cloud shadows pixels. This band was included in the S2 dataset as an additional band.

4.1.2 Sentinel-3 OLCI Level-1B Product

S3 OLCI products are also available in two levels of processing for the users: Level-1B (L1B) radiance product and Level-2 land, water and atmospheric geophysical parameter products [8]. All the products are available in either full or reduced version with a spatial resolution at 300 m and 1200 m respectively. L1B products are radiometrically corrected, ortho-geo-located in WGS84 geographic projection and contain Top-Of-Atmosphere (TOA) radiometric measurements. Level-2 products are higher level geophysical products derived from L1B with additional atmospheric and other corrections applied. This study uses L1B full resolution product named as OL_1_EFR to utilize the information contained in full spectral ranges in 21 bands. The S3 user products are available in Product Dissemination Units (PDU), defined as a "frame" for S3 OLCI L1B full resolution products. These products are available at Copernicus Open Access Hub in file format based on SAFE with the image data in NetCDF 4 format.

The band characteristics of OLCI sensor are listed in the table 4.2. The spectral bands range from the visible to the near-infra-red (390 nm to 1040 nm) and serve the various application purposes. Bands Oa1-10 are in visible spectral domain, Oa11 in red edge, Oa12–16 are in oxygen absorption region, Oa17–19 are located further in the NIR bands, Oa19–20 in atmospheric water absorption features and the last band Oa21 in the SWIR region.

The dataset for this thesis consists of S3 images of the same day as corresponding S2 product with a maximum sensing time difference of an hour. The TOA radiance values in S3 has to be converted to reflectance values using the available information on sun zenith angle and solar flux value for further processing. Radiance is the brightness from the target directly measured by remote sensing instruments whereas reflectance is the ratio of the amount of light leaving a target to that striking the target. Reflectance is a dimensionless quantity and is independent of the intensity and nature of illumination. Thus, it is necessary to convert radiance to reflectance as it provides a standardized measure which is directly comparable between images from different sensors, locations, and times [69]. This conversion is done by applying Rayleigh correction for molecular scattering since the complete atmospheric correction similar to S2 is not yet available for S3 [11]. Cloud mask is also not provided with the product so it was generated during pre-processing.

Table 4.2: S3 OLCI spectral bands definition, Source: [8]

Band	Central wavelength (nm)	Bandwidth (nm)	Spatial Resolution (m)	Description
Oa1	400	15	300	Aerosol correction, improved water constituent retrieval
Oa2	412.5	10	300	Yellow substance and detrital pigments (Turbidity)
Oa3	442.5	10	300	Chlorophyll (Chl) absorption max., biogeochemistry, vegetation
Oa4	442	10	300	High Chl, other pigments
Oa5	510	10	300	Chl, sediment, turbidity, red tide
Oa6	560	10	300	Chl reference (Chl minimum)
Oa7	620	10	300	Sediment loading
Oa8	665	10	300	Chl (2nd Chl abs. max.), sediment, yellow substance/vegetation
Oa9	673.75	7.5	300	For improved fluorescence retrieval
Oa10	681.25	7.5	300	Chl fluorescence peak, red edge
Oa11	708.75	10	300	Chl fluorescence baseline, red edge transition
Oa12	753.75	7.5	300	O2 absorption/clouds, vegetation
Oa13	761.25	2.5	300	O2 absorption band/aerosol corr.
Oa14	764.38	3.75	300	Atmospheric correction
Oa15	767.5	2.5	300	O2A used for cloud top pressure, fluorescence over land
Oa16	778.75	15	300	Atmospheric corr./aerosol corr.
Oa17	865	20	300	Atmospheric corr./aerosol corr., clouds, pixel co-registration
Oa18	885	10	300	Water vapour absorption reference band, vegetation monitoring.
Oa19	900	10	300	Water vapour absorption/vegetation monitoring (max. reflectance)
Oa20	940	20	300	Water vapour absorption, atmos./aerosol corr.
Oa21	1020	40	300	Atmospheric/aerosol corr.

4.2 S2/S3 Paired Dataset Preparation

Preparation of a paired S2/S3 dataset is an important as well as time consuming step of this research. The process of preparing the dataset is explained in the following subsections.

4.2.1 Data Download

The first step in dataset preparation is collecting the corresponding S2 L2A and S3 OL_1_EFR products over the study area for the year 2019 from Copernicus Open Access Hub. An initial query on the Copernicus Open Access Hub showed that there were 77 available products for S2 and 395 for S3 in the given area and time frame. To avoid the

tedious task of manually selecting and downloading, automation was implemented for data acquisition using the available Copernicus data provision services. The Sentinelsat [2] API was implemented in a Python environment for query and download as well. First, the extent of S2 *30STJ* tile was defined as our ROI with the corner coordinates in the GeoJSON format. Then, a query was formed by defining the platform name, the product type and the temporal interval. In case of S2, the returned products were filtered by *tileid* whereas for S3 an area relation was defined to select all the S3 products that contain the ROI. After obtaining the list of products, an iteration was applied to download all the products automatically. However, a slightly different approach had to be applied for Long Term Archive (LTA) products, the products archived by the data hub according to their rolling plan that starts with the oldest data. At the time of data downloading, S2 products before 1 July, 2019 and S3 products before 1 September, 2019 were in LTA. So these offline products had to be retrieved from the archive first. Sending a download request triggered the product retrieval and they could be downloaded after being fully restored. The restored products are online for a limited amount of time and can be downloaded by any user within that duration. The products were renamed using the sensor name and the day of the year, also called the Julian days. For example: the S2 and S3 images captured on January 12, that is day 12 are named as s2_012 and s3_012 respectively.

4.2.2 Data Pre-processing

Next, the data pre-processing was conducted using Sentinel Application Platform (SNAP), a common software platform developed by ESA for processing, analyzing and visualizing the earth observation data from all the Sentinel missions as well as other missions. To handle the automatic processing of large amount of data, it provides the following two options: Graph Processing Framework (GPF) for creating processing chains or the SNAP-Python interface, also known as *"snappy"* to access the SNAP Java API from Python. The latter was chosen for our work because of its flexibility to be customised and reused according to user needs. The pre-processing is presented graphically in the figure 4.2. Two separate pipelines were developed for processing S2 and S3 products. S2 products were first resampled to 20 m considering the data volume and also because any further processing required all the bands to be in the same spatial resolution. Then a spectral subset was applied to retain 12 spectral bands plus 1 scene classification band in the final products.

Regarding S3 products, first spatial subset was applied to limit the further processing to our area of interest. Then, Rayleigh correction was applied to the subsets resulting in Bottom of Rayleigh corrected Reflectance (BRR) product. The same spatial subsets were processed through IdePix OLCI pixel identification tool [70] to obtain a pixel classification band with clouds and cloud shadows pixels identified. This band was merged with Rayleigh corrected products before further processing.

[2] https://github.com/sentinelsat/sentinelsat/

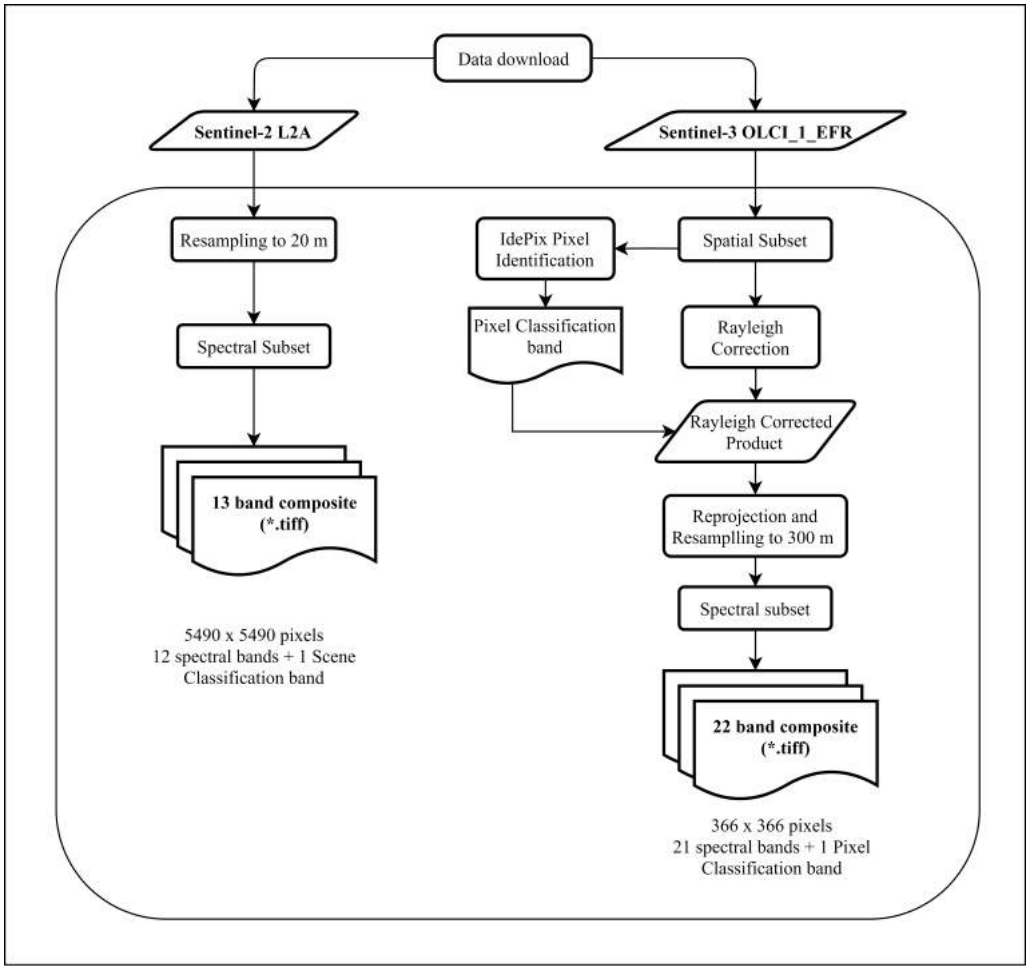


Figure 4.2: S2 and S3 data pre-processing

Then, the merged products were reprojected from geographic WGS84 coordinate system to the projected UTM 30 N system to co-register with S2. During reprojection, the resampling of the products at 300m resolution was performed. After that, a spectral subset was done to retain 21 reflectance bands plus 1 pixel classification band in each final product. Finally, the co-registered image composites are obtained in the GeoTIFF file format with S2 of size $5490 \times 5490 \times 13$ and $366 \times 366 \times 22$ for S3. During the pre-processing, S3 products are read with pixel based geocoding to ensure high level of positioning accuracy and exact overlay with S2 images.

4.2.3 Sentinel-2 Data Products Generation

S2 images are stored in their original size of $5490 \times 5490 \times 13$ pixels in order to maintain the flexibility to create several data products with different band combinations. The scene classification and pixel classification bands are used as masks to filter clouds and invalid pixels for S2 and S3 respectively. S2 is resized to S3 spatial resolution and then the data products NDVI and SAVI are generated during the data extraction stage (details in section 5.3.1) inside prediction models. The mathematical formula to derive them is given in section 3.1 of chapter 3.

4.3 Resources Used

Many open source software and packages have been used in this research. The mostly used platforms are highlighted here.

Python is an object-oriented, interactive and open-source high-level programming language that consists of modules, classes, libraries and interfaces for easy programming in different operating systems with dynamic typing. The project uses Python 3.6 version.

Tensorflow is a Google Brain team created open-source library offering high computational tasks like machine learning and deep learning through usages of computational platforms like CPUs, GPUs, and TPUs. It uses Python as a front-end API. The research uses Tensorflow 1.12.0 version for the project.

Keras is a user-friendly, high-level python written API that supports easy and faster deep learning neural networks with minimum coding running on top of the machine learning platform, TensorFlow. It offers a complete framework for successfully building and running a neural network model. The research is using Keras 2.2.4 version.

Within these main frameworks and programming platforms, other modules, libraries, and packages supported by python were used for data download, pre-processing, image visualization, developing networks and plots. These include Sentinelsat and SNAP API, numpy, scikit-learn, rasterio, opencv, and matplotlib. In terms of hardware, the experiments in this study were carried out on a linux server with GeForce RTX 2080 Ti GPU (11GB RAM).

5 METHODOLOGICAL DESCRIPTION

This chapter discusses the implementation methodology of the selected algorithms and their performance evaluation. At the beginning, a description of the network architectures of the deep learning and machine learning models are provided, followed by explanations on the implementation aspects of training, experimental settings, and quantitative and qualitative performance evaluation.

5.1 Deep Learning Network Architectures

For the purpose of vegetation products estimation, two types of networks are chosen and employed in the order of increasing depth and complexity to analyze the performance as network architectures change. First is a simple multilayer perceptron (MLP) architecture, and the second is state of the art Convolutional Neural Networks (CNN).

5.1.1 Multilayer Perceptron

A simple MLP with a single hidden layer (figure 5.1), referred hereafter as MLP, was adapted from the study by [43] where it has been used for classification task using hyperspectral imagery. The units in input layer corresponds to the number of samples supplied and the hidden layer is a FC layer with $n_{bands} \cdot \frac{2}{3} + 10$ number of units and ReLU activation. The output is also a FC layer having linear activation with a single unit output of the estimated vegetation product value. MLP has been implemented in two ways to enable pixel-wise and patch-wise inputs and to introduce change in the complexity of the model.

MLP-1D takes pixel-wise input of central pixel value across all S3 spectral bands from a given patch and the corresponding pixel value in S2 vegetation product value. Each unit of input layer is a vector of length 21.

MLP-2D also has the same architecture as MLP-1D, the only difference is input and the number of units in the hidden layer. This model takes patch-wise input, that consists of all pixel values across all of the S3 spectral bands within a patch size of $p \times p$ and the corresponding central pixel value of the patch in S2 vegetation product value.

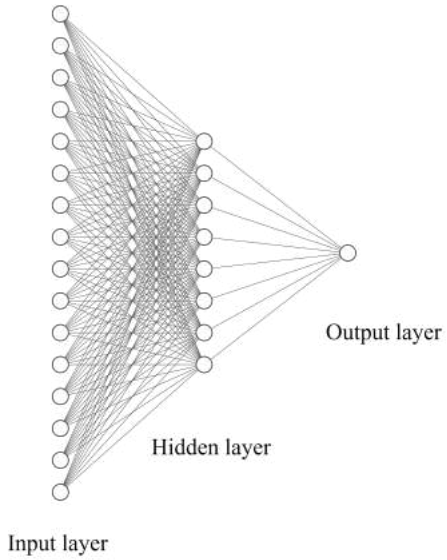


Figure 5.1: Architecture of Multilayer Perceptron (MLP)

Even though the pixel values are in a matrix form, they are flattened to one dimensional vector before feeding to the network. Thus, the each input is a vector of length $p \times p \times 21$. With the increase in patch size, this number and subsequently the number of units in the hidden layer as well as the parameters to be trained can increase exponentially making this model more complicated than MLP-1D.

5.1.2 Convolutional Neural Network

Among the four CNN architectures used in this study, the first one is a basic 2D CNN architecture based on the classical LeNet-5 template, referred as CNN2D-1 while the second one was adapted from [45] referred as CNN2D-2. The other two architectures are 3D implementations of the same 2D CNNs, CNN2D-1 and CNN2D-2 and name CNN3D-1 and CNN3D-2 respectively. The input for all 2D CNN architectures is S3 patches of size (p, p, 21). 3D CNNs have an additional temporal dimension which is determined by the depth. A depth value of 1 is considered so that each training input consists of S3 patches extracted from 3 consecutive images (current image, 1 before and 1 after) and the input size is (p, p, 3, 21).

CNN2D-1

The first 2D CNN model, CNN2D-1 consists of 4 layers as shown in figure 5.2. The convolution, activation and pooling are combined and considered one layer. This combination is in initial two layers of this network with 64, and 128 filters respectively. The number of filters is increased by a factor of 2 with each subsequent convolutional layer and a kernel size of 3×3 is used for both layers. The chosen activation function is rectified non-linear activation function (ReLU). Then, downsampling is done by a max-pooling layer of size 2×2 and the spatial dimension of the input is reduced by half. In the first FC layer, each neuron provides a full connection to all the learned feature maps issued from the previous

layers. This FC layer uses ReLU activation. Then the last FC layer with linear activation at the end with 1 neuron predicts the value of output.

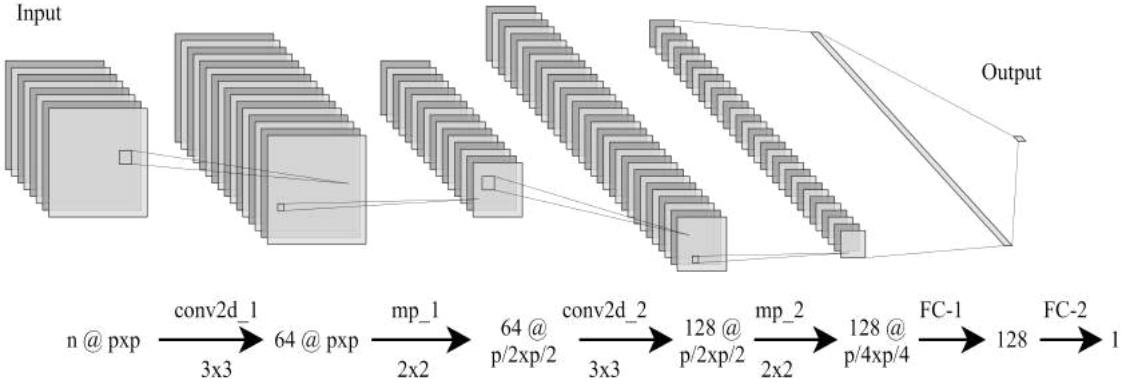


Figure 5.2: Architecture of CNN2D-1 (n: number of bands, p: patch size, conv2d_j: j^{th} 2D convolution layer, mp_j: j^{th} Max-pooling layer, FC_j: j^{th} fully connected layer)

CNN2D-2

CNN2D-2 network architecture (figure 5.3) was created based on the works of [45]. The authors developed this patch based CNN network specifically for the classification of medium resolution TOA reflectance data. This architecture was chosen since S3 also falls under the category of medium resolution image. The major difference from the previous CNN2D-1 is that this network comprises of 5 convolutional layers with no pooling in between and a final FC layer followed by softmax layer which was changed to another FC with linear activation and 1 neuron to fit this study of regression.

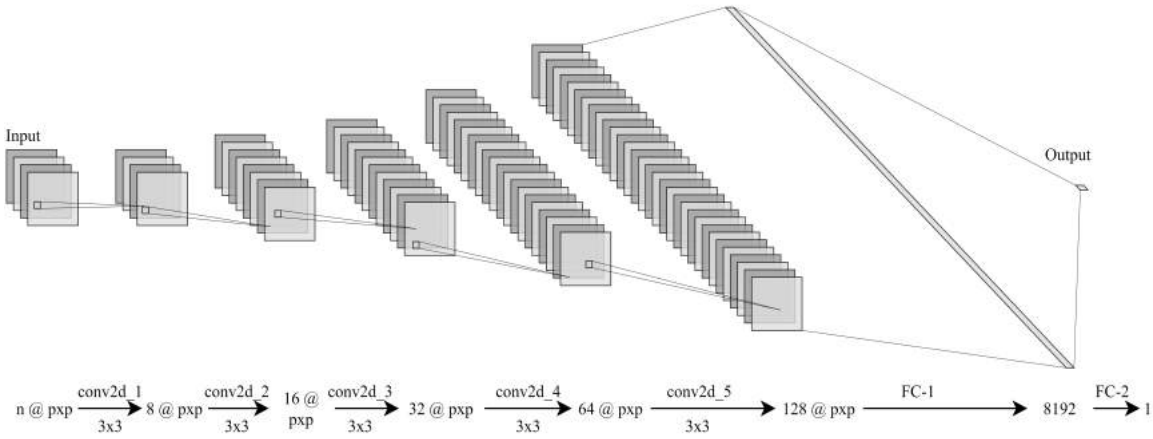


Figure 5.3: Architecture of CNN2D-2 (n: number of bands, p: patch size, conv2d_j: j^{th} 2D convolution layer, FC_j: j^{th} fully connected layer)

The filters are of size 8, 16, 32, 64 and 128 with 3×3 kernels and the first FC layer has 8192 neurons which is equivalent to $128 \times p \times p$ for the patch size of 8. This is in consistency with the FC layer in [45] architecture that has 3200 neurons for patch size of 5. The high number of neurons in this FC layer increases the parameters for this network exponentially. Finally, the last FC layer is a 1 neuron layer with linear activation that predicts the results, similar to the previous architecture.

CNN3D-1

The network architecture of CNN3D-1 is the same as CNN2D-1 with two convolution-max pooling blocks and two fully connected layers. The only exception is an additional temporal dimension and all the 2D convolution and pooling layers being replaced by 3D. The filters of size 64 and 128 are used for the first and second convolutional layers and the kernel size, 3×3×3, is used for both layers. The layers architecture is presented in table 5.1.

Table 5.1: CNN3D-1 layers architecture

Layer name	Operation	Filters	Kernel size	Output Shape
conv3d_1	Conv3D	64	3×3×3	8×8×3×64
max_pooling3d_1	MaxPooling3D	-	2×2×2	4×4×2×64
conv3d_2	Conv3D	128	3×3×3	4×4×2×128
max_pooling3d_2	MaxPooling3D	-	2×2×2	2×2×1×128
FC-1	Fully connected	-	-	1×128
FC-2	Fully connected	-	-	1

CNN3D-2

The network architecture of CNN3D-2 is similar to CNN2D-2 with an additional temporal dimension and all the 2D convolution and pooling layers replaced by 3D. The filters of size 8, 16, 32, 64 and 128 are used in the five convolutional layers with 3×3×3 kernels. The major difference from the previous CNN3D-1 is that no pooling layers are used and with same padding the spatial and temporal dimension is maintained through all the successive layers until the fully connected layers. The layers architecture is presented in table 5.2.

Table 5.2: CNN3D-2 layers architecture

Layer name	Operation	Filters	Kernel size	Output Shape
conv3d_1	Conv3D	8	3×3×3	8×8×3×8
conv3d_2	Conv3D	16	3×3×3	8×8×3×16
conv3d_3	Conv3D	32	3×3×3	8×8×3×32
conv3d_4	Conv3D	64	3×3×3	8×8×3×64
conv3d_5	Conv3D	128	3×3×3	8×8×3×128
FC-1	Fully connected	-	-	1×8192
FC-2	Fully connected	-	-	1

5.2 Classical Regression Models

Four classical machine learning regression algorithms with increasing complexity and ability in solving linear to non-linear regression problems have been considered for the purpose of this study namely linear regression, ridge regression, SVR and RFR. Among the four algorithms, linear and ridge regression are linear based methods, ridge regression being an extension of linear with added L2 regularization. SVR with non-linear radial basis function (RBF) kernels and RFR are non-linear regression algorithms. Each model implementation has two variations based on the type of input either pixel-wise or patch-wise, similar to the MLP models. 1D ML models take the pixel-wise input of central pixel value across the S3 spectral bands from a given patch and the corresponding pixel value in S2 vegetation product value. 2D ML models take patch-wise input of all pixel values across the S3 spectral bands within a patch size of $p \times p$ and the corresponding central pixel value of the patch in S2 vegetation product value. The values in matrix like form are flattened to one dimensional vector of length $p \times p \times 21$ before feeding to the model.

5.3 Model Training

5.3.1 Data Extraction and Normalization

Before training the ML and DL models, it is essential to prepare the dataset so that the information contained within the dataset is best exposed to the network. The dataset available after the pre-processing steps explained in chapter 4 consists of 41 pairs of S3 images of size $366 \times 366 \times 22$ and S2 images of size $5490 \times 5490 \times 13$. Firstly, the last bands of S3 and S2, the pixel classification and scene classification bands are separated from the image composite and used as masks over the remaining spectral bands to convert clouds, cloud shadows or invalid pixels to no-data. The no-data values are filtered during patch extraction. Since the S3 and S2 images have different spatial resolutions, S2 is then resized to S3 resolution of 300 m. After that, S2 vegetation products (NDVI or GCI) are calculated when running the models for each product considered. Among 41 image pairs, 2 are kept separate for qualitative evaluation in the end. The samples are collected only from 39 image pairs as patches of a defined size with spectral values across all 21 bands from S3 (input features) and corresponding central pixel value in S2 product (target/ground truth labels). For 3D CNN models, the sample consists of three dimensional S3 patches extracted from 3 consecutive images (current image, 1 before and 1 after) when depth is set to 1 and corresponding central pixel value in S2 product.

Considering that the no-data pixels might affect the number of samples extracted for different patch sizes considered in hyperparameter tuning of DL models and for comparison in ML models, a smaller dataset was created consisting of 17 cloud free images from among the 41 images excluding the tiles separated for qualitative evaluation. No masking was applied in this dataset to keep the samples consistent throughout the varying patch sizes for comparability. Hyperparameter tuning was done using this dataset to find the best configurations for each DL model and then they were trained and tested

on the whole dataset. This dataset is also used to analyze the effects of patch size in 2D ML models.

The samples from both dataset are split into training, validation and test dataset. Then, a min-max normalization technique is applied to rescale the data to the range of [0,1] before training the models.

5.3.2 Training

The general training process of deep learning model consists of forward computation, loss optimization, and back-propagation and parameter updating. The input S3 patches/values with corresponding S2 vegetation product values, which are our ground truth labels, are input to the prediction models as the training samples. Within the MLP architecture, the neurons transmit the input values to the hidden layer where an activation is performed on the weighted combination of the inputs to calculate and forward the outputs. For CNN models, convolution and pooling layers extract important features from the input images and forward it to succeeding layers. Then, the first FC layer in network collects all the outputs from the previous layers as a flattened array and the last fully connected layer predicts the S2 data product value corresponding to the input S3 patch. Then the chosen loss function, mean squared loss, is computed for both MLP and CNN models by comparing predicted output against the ground truth values. This loss is back-propagated to update the network parameters using Adam optimization. The machine learning regression models are also fit with the same training data as deep learning models using the scikit-learn library of Python. Linear regression and ridge regression fit a linear model, SVR with non-linear radial basis function (RBF) kernels and RFR fit non-linear models. After fitting, the models learn from the training data and can be used to predict the output using test data.

5.4 Experimental Settings

The experiments carried out in this research are built on top of deep learning library "Keras" with “Tensorflow” backend in python environment. The input shape for the various ML and DL models differ according to whether they are pixel-wise, patch-wise or patch-based as explained in previous sections. To summarize, ML and MLP models are pixel-wise and patch-wise while the CNN models are patch-based. Samples are extracted as 2D/3D patches from S3 and corresponding central pixel values from S2. They are split into training and test data (60% - 40%), 20% of training data is used for validation in deep learning models. The networks are trained using Adam optimizer for 100 epochs. Early stopping with a patience value of 40 is incorporated into the model based on the validation dataset in order to avoid the overfitting. L2 regularization is used in all CNN models and its value was set as 0.0001.

Experiment 1: Hyperparameter optimization

In the first experiment, hyperparameter optimization of DL models is done for their best performance. The hyperparameters that are tested in this experiment are patch size, batch size, and learning rate. Previous similar studies and theoretical background behind each hyper-parameters were taken as the basis behind choosing the values to be tested. While testing, only one hyperparameter is changed at one time keeping all others constant. The test was performed on all six DL models for predicting both NDVI and SAVI products. The evaluation was done based on mean squared error.

Table 5.3: Hyperparameters to be optimized and the tested values

Hyperparameters	Tested Values
Patch Size	8, 16, 32
Batch size	64, 128
Learning rate	0.0001, 0.001, 0.01

Experiment 2: Comparing pixel-wise and patch-wise models

All ML models and MLP model are implemented with pixel-wise and patch-wise inputs. This experiment is done to analyze whether including the neighbouring pixel information instead of single pixel improves the model performance. The effect of varying patch sizes on patch-wise ML models is also studied.

Experiment 3: Simplifying CNN2D-2 and CNN3D-2 architectures

In the third experiment, the network architectures of CNN2D-2 and CNN3D-2 are modified by reducing the number of neurons on the first FC layer. Since the large number of neurons on the first FC layer made the networks computationally costly, this experiment is done to analyze the effect of varying this layer on the performance of the models. For this experiment, the first FC layer with 8192 neurons is replaced by a FC layer with 128 neurons, equal to the number of filters in the previous layer for both CNN2D-2 and CNN3D-2, keeping everything else constant.

5.5 Performance Evaluation

During hyperparameter tuning experiments, the performance of the deep learning models is evaluated using MSE metric. After these experiments, the deep learning models with the best hyperparameter combinations are selected for final training and prediction from the input dataset. The results from all ML regression models and deep learning models are then quantitatively evaluated based on MSE, RMSE and MAE and qualitatively through the visual inspection of prediction maps generated for new image tiles the models have not seen yet.

5.5.1 Quantitative Evaluation

For quantitative evaluation, standard metrics based on the problem are used. Three metrics, MSE, RMSE and MAE are used to evaluate the performance of the regression models.

Mean squared error (MSE) is the average of squared differences between the actual and the predicted values. It is a metric often used in regression problems due to its simplicity and quadratic error computation, which penalizes predictions that substantially differ from the corresponding reference values. The predictions will be more accurate when the value of MSE is lower and close to 0 which denotes smaller differences from the reference. The mathematical formulation to calculate MSE is given in the equation:

$$MSE = \frac{1}{n}\sum_{j}^{n}\left(y_j - \widehat{y_j}\right)^2 \tag{5.1}$$

where, y_j is the actual value, $\widehat{y_j}$ is the predicted value and n is the number of samples.

Root Mean squared error (RMSE) is the standard deviation of the errors which occur when a prediction is made on a dataset. Lower the value of RMSE, better is the performance of the model. The mathematical formulation to calculate RMSE is given in the equation:

$$RMSE = \sqrt{\frac{1}{n}\sum_{j}^{n}\left(y_j - \widehat{y_j}\right)^2} \tag{5.2}$$

Mean absolute error (MAE) measures the average magnitude of the errors in a set of predictions without considering their direction. It is the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight. The mathematical formulation to calculate MAE is given in the equation:

$$MAE = \frac{1}{n}\sum_{j}^{n}\left|y_j - \widehat{y_j}\right| \tag{5.3}$$

5.5.2 Qualitative Evaluation

For qualitative evaluation, 2 pairs of S2/S3 image which have not been included in the train-test dataset are used. After hyperparameter tuning and obtaining the quantitative results from all the experiments, the deep learning models with the best hyperparameter combinations and ML models are used to generate the vegetation product (NDVI and SAVI) maps for those 2 images. These maps are evaluated against the vegetation product map directly calculated from S2 image which will be our ground truth map. The absolute error maps are also computed for each predicted map that help to visualize the differences between the ground truth and predicted values.

6 RESULTS AND DISCUSSION

This chapter presents the results obtained in the experimental designs and performance evaluation as detailed in Chapter 5. First, the findings of the experiments are given succeeded by an explanation of the performance evaluation of the regression models. The vegetation product estimation maps generated by all the implemented models are presented in the last section.

6.1 Experiment Results

6.1.1 Deep Learning Models Hyperparameter Optimization

The results of the hyperparameter tuning tests performed in predicting NDVI products are explained in the following sections.

6.1.1.1 Patch Size

The effect of varying sample patch size on the deep learning models is shown in figure 6.1. The value of learning rate is set to 0.001 with batch size 64 during this experiment. Due to the limitations in computation capacity, the test could not be completed for CNN2D-2 model for patch size 32 and CNN3D-2 for patch size 16 and 32. The value plotted is MSE, so a lower value means better performance of the model.

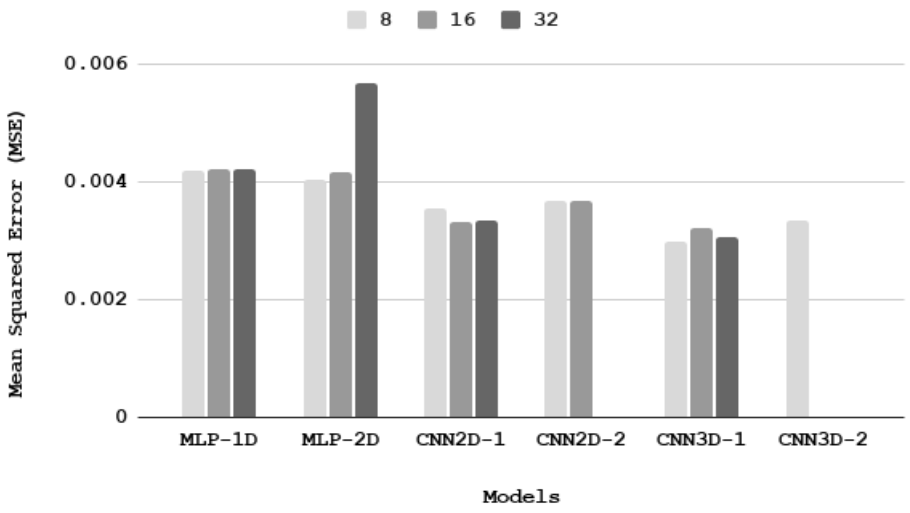


Figure 6.1: Effect of varying sample patch size

Even though studies have shown that larger patch sizes are considered better in representing spatial context in CNNs, the models in this study did not necessarily conform to this. Out of the six models, only CNN2D-1 showed decrease in MSE with increase in patch size. The input images being mid resolution at 300 m pixel size could be one reason for this [45]. Here, MLP-1D is a pixel-wise model so the varying patch size has no effect on this model. Considering the results as well as computational capacity, all the remaining experiments are performed with input patch size of 8.

6.1.1.2 Batch Size

Figure 6.2 depicts the change in the performance of the models with the varying batch size. The learning rate value is set to 0.001 with patch size 8 during this experiment. The figure shows all of the models performed better with the lower batch size of 64 with the exception of CNN2D-1, for which there was no effect seen. According to [71], larger batch sizes tend to generalize poorly causing loss of the model to increase. Thus the batch size of 64 was fixed for all the models.

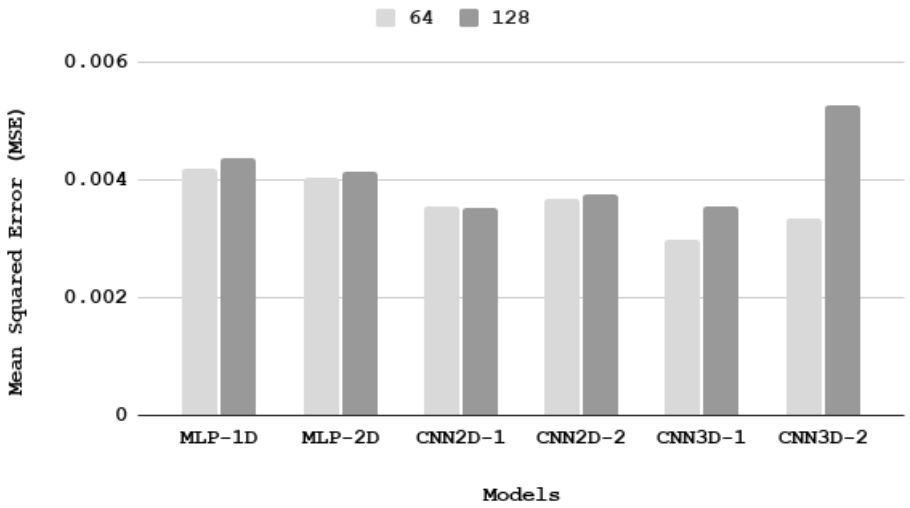


Figure 6.2: Effect of varying batch size

6.1.1.3 Learning Rate

The effect of varying learning rate on the deep learning models is shown in figure 6.3. The patch size is set to 8 with batch size 64 during this experiment. The results of varying learning rates indicate that the more complex networks CNN2D-2 and CNN3D-2 perform better with a smaller learning rate of 0.0001 while for others the learning rate of 0.001 worked well. On the basis of the results, the learning rate of 0.0001 is set for CNN2D-2 and CNN3D-2, which is the learning rate suggested in [45], from where the model is adapted and the learning rate of 0.001 is used for the rest of the models.

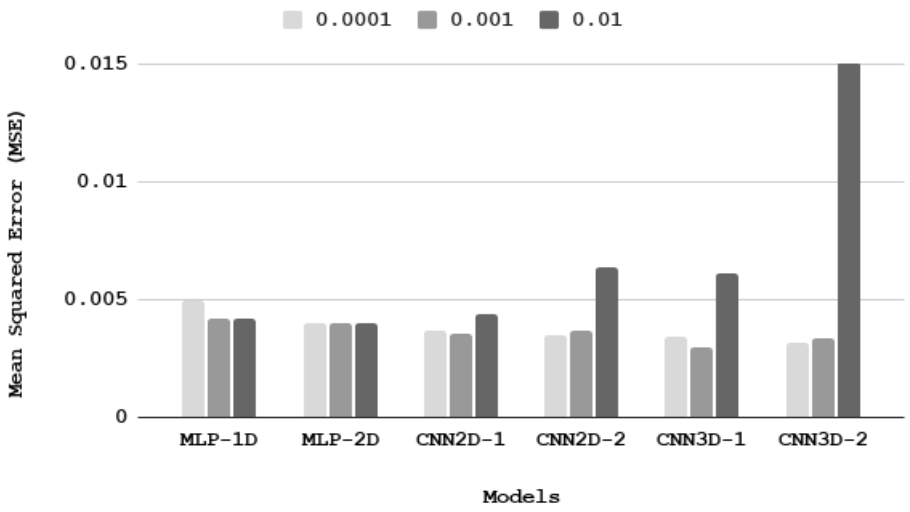
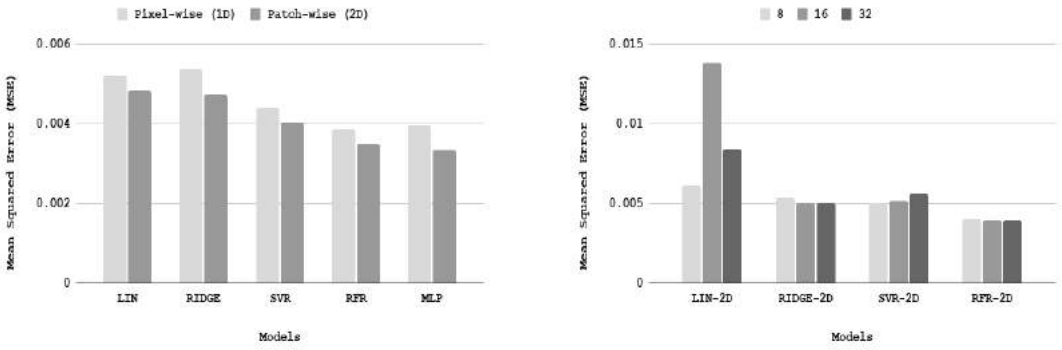


Figure 6.3: Effect of varying learning rate

The hyperparameter tuning tests were performed for both NDVI and SAVI however the results from only NDVI are reported here as the results were similar for both. The same hyperparameters were adopted during SAVI prediction as well.

6.1.2 Comparing pixel-wise and patch-wise models

The effects of varying the type of input and patch sizes in the DL and ML models for NDVI product prediction are presented in figure 6.4. All the patch-wise DL and ML models (patch size - 8), performed better than their pixel-wise counter-parts with lower MSE values. This result is consistent with the study by [72], where patch-wise ML models outperformed pixel-wise ones in classification task.



(a) Varying input type

(b) Varying patch sizes

Figure 6.4: Effect of varying input type and patch sizes

However, in case of varying patch sizes in patch-wise (2D) ML models, a mixed result was obtained similar to the result for DL models shown in section 6.1.1.1. In the figure, LIN refers to linear, and RIDGE refers to ridge regression. RIDGE and RFR showed

improvement in performance with increasing patch size whereas LIN and SVR performed poorly. The patch size of 8 was used for the generation of final results.

6.1.3 Simplifying CNN2D-2 and CNN3D-2 architectures

The first FC layer in CNN2D-2 and CNN3D-2 consist of 8192 neurons. One of the major drawbacks of such FC layer is that since all the neurons are fully connected, the number of parameters can grow to very high with the increase in the number of neurons. So for this experiment, the first FC layer with 8192 neurons is replaced by a FC layer with 128 neurons, equal to the number of filters in the previous layer for both CNN2D-2 and CNN3D-2, keeping everything else constant. These modified models will be referred to as CNN2D-2-1 and CNN3D-2-1 hereafter. The table 6.1 shows the number of parameters and MSE obtained in NDVI prediction for each of the deep learning models.

Table 6.1: Number of parameters of deep learning models

Model	No. of Parameters	MSE
MLP-1D	553	0.00396
MLP-2D	1,219,477	0.00334
CNN2D-1	151,809	0.00278
CNN2D-2	67,224,929	0.00252
CNN2D-2-1	1,148,513	0.00242
CNN3D-1	323,457	0.00214
CNN3D-2	201,641,521	0.00225
CNN3D-2-1	3,444,529	0.00198

Even with very less parameters than CNN2D-2 and CNN3D-2, it is evident that CNN2D-2-1 and CNN3D-2-1 perform very well in terms of quantitative evaluation by MSE metric.

6.2 Performance Evaluation

After the completion of hyperparameter optimization experiments, each DL model with the best hyperparameter combination was implemented for the final training and prediction of vegetation product maps for NDVI and SAVI along with the ML models. Three metrics, RMSE, MSE and MAE were computed for all the models to evaluate their performance. The final models were run on the samples extracted from the dataset with 39 images. Among the samples, 60% was used for training while 40% was reserved for testing purpose. Table 6.2 and 6.3 present the quantitative evaluation of the predicted vegetation indices, NDVI and SAVI in terms of RMSE, MSE and MAE metrics. The results for all of the models have been tabulated where the "Input" column indicates the type of input fed to the model, that is pixel-wise, patch-wise or patch-based. In the model column LIN refers to linear, and RIDGE refers to ridge regression. Other notations have been introduced in the previous chapters. From the table, it is clearly seen that all the DL models except

pixel-wise MLP (1D) outperformed the classical ML regression approaches for both NDVI and SAVI prediction.

Table 6.2: Prediction results for NDVI in terms of RMSE, MSE and MAE

NDVI				
Input	**Model**	**RMSE**	**MSE**	**MAE**
Pixel-wise (1D)	LIN	0.07230	0.00523	0.04714
	RIDGE	0.07327	0.00537	0.04703
	SVR	0.06624	0.00439	0.04932
	RFR	0.06216	0.00386	0.03966
	MLP	0.06295	0.00396	0.04127
Patch-wise (2D)	LIN	0.06953	0.00483	0.04798
	RIDGE	0.06868	0.00472	0.04582
	SVR	0.06355	0.00404	0.04789
	RFR	0.05915	0.00350	0.03836
	MLP	0.05782	0.00334	0.03857
Patch-based (2D)	CNN2D-1	0.05270	0.00278	0.03509
	CNN2D-2	0.05024	0.00252	0.03377
	CNN2D-2-1	0.04919	0.00242	0.03204
Patch-based (3D)	CNN3D-1	0.04626	0.00214	0.03109
	CNN3D-2	0.04747	0.00225	0.03207
	CNN3D-2-1	**0.04449**	**0.00198**	**0.02957**

Table 6.3: Prediction results for SAVI in terms of RMSE, MSE and MAE

SAVI				
Input	**Model**	**RMSE**	**MSE**	**MAE**
Pixel-wise (1D)	LIN	0.08210	0.00674	0.05602
	RIDGE	0.08248	0.00680	0.05646
	SVR	0.08081	0.00653	0.05635
	RFR	0.08161	0.00666	0.05396
	MLP	0.07991	0.00639	0.05317
Patch-wise (2D)	LIN	0.08071	0.00651	0.05688
	RIDGE	0.07899	0.00624	0.05444
	SVR	0.07655	0.00586	0.05588
	RFR	0.07845	0.00616	0.05215
	MLP	0.07557	0.00571	0.05104
Patch-based (2D)	CNN2D-1	0.07151	0.00511	0.04858
	CNN2D-2	0.07062	0.00499	0.04777
	CNN2D-2-1	0.06666	0.00444	0.04397
Patch-based (3D)	CNN3D-1	0.06006	0.00361	0.04121
	CNN3D-2	0.05668	0.00321	0.03803
	CNN3D-2-1	**0.05310**	**0.00282**	**0.03556**

CNN3D-2-1 is the best performing model among all, with the lowest values in all three error metrics for both NDVI and SAVI. It has a remarkable MSE of 0.00198 when compared to the best performing ML algorithm, patch-wise RFR (2D) with MSE of 0.0035 in case of NDVI. For SAVI, SVR-2D is the best performing ML algorithm with MSE of 0.00586 while CNN3D-2-1 has MSE 0.00282. The values for other metrics, RMSE and MAE

are lower for CNN3D-2-1 in similar proportion. Among the 2D CNN models, CNN2D-2-1 has the lowest error metrics for NDVI. The linear regression algorithms, LIN and RIDGE are the worst performing ones. These observations are similar in case of SAVI as well.

Overall, it is seen that the non-linear ML methods perform better than the linear ones. Having patch-wise inputs help to improve them further. However, DL models outperform them with CNN being superior over MLP with simple architecture which was not able to learn the features as much as CNN. Increasing the layers in 2D CNN improved the results from CNN2D-1 to CNN2D-2 and the best result in 2D CNN was achieved by decreasing the large number of neurons which often lead to overfitting. Implementing 3D CNN with temporal information as additional dimension helped to achieve the best results. Even though there are not similar literature available for comparison, the findings of this study are consistent with the works done for yield estimation [42] [16] and classification [72] tasks which show that the CNN architectures outperform the ML regression algorithms and 3D CNN are able to give the best performance of all.

6.3 Comparison of Predicted Vegetation Product Maps

The vegetation product maps generated by all the implemented models for NDVI and the corresponding ground-truth S2 NDVI map are shown in figure 6.5. The maps have been normalized in the range [-1,1] to assist in visual comparison. Even though all the maps look visually similar to the ground truth, differences can be seen in the places of heterogeneous pixels. For example, the river pixels on the bottom right corner are not predicted correctly by many methods. In the LIN and RIDGE maps, the river is not visible while for MLP, RFR and SVR maps, the pixels are darker than the ground truth.

To better visualize the differences in predictions, absolute error maps were also computed for each predicted map as shown in figure 6.6. The maps have been normalized from [0,2] and the color scheme goes from light to dark with darker values corresponding to higher error. The absolute error maps also depict that the higher differences are in the places of heterogeneous pixels specially near the water bodies and the central high vegetation region with built up area.

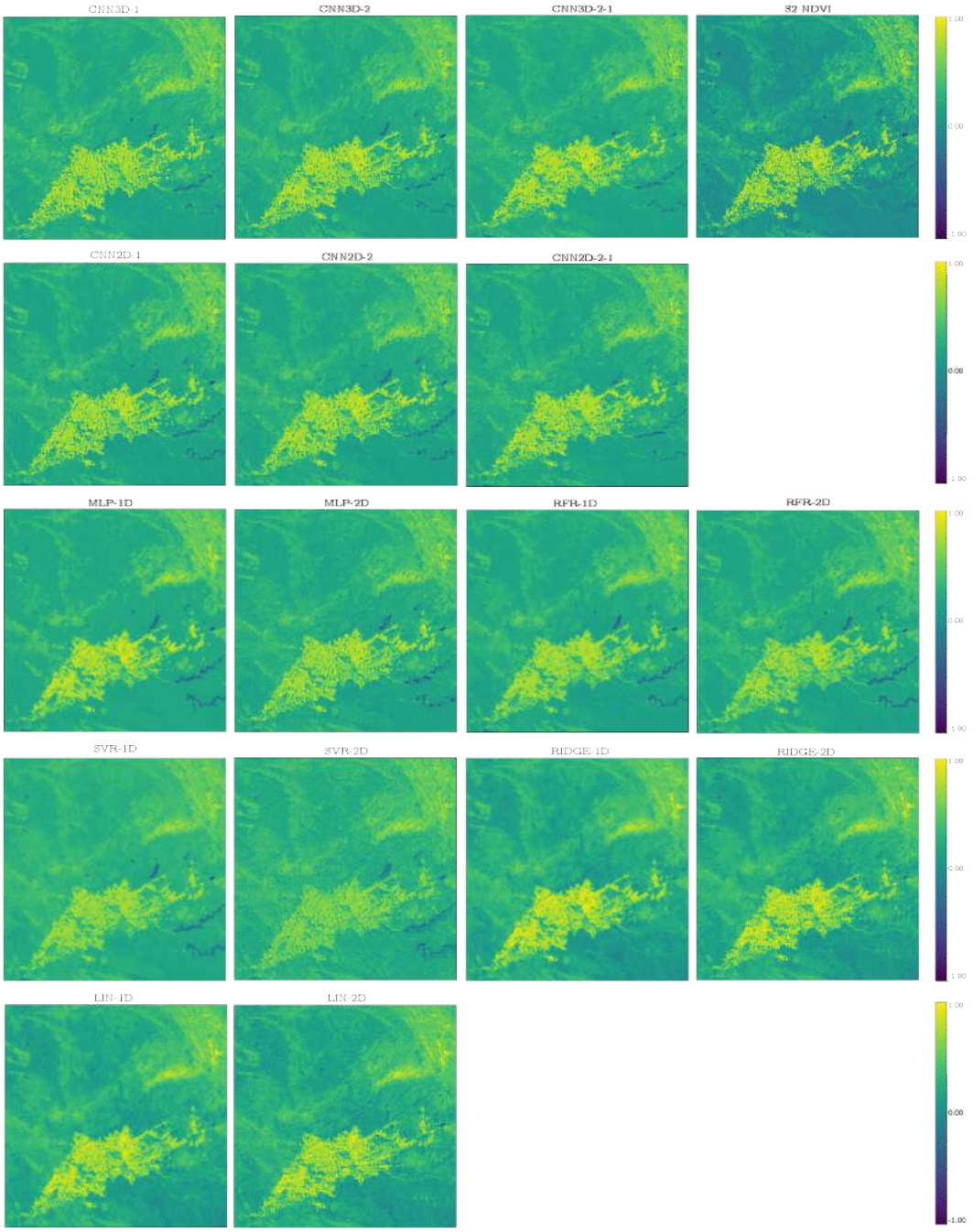


Figure 6.5: Ground truth S2 NDVI map on 11 July, 2019 on top right corner; NDVI maps generated by the prediction models (the scale from dark to light represents the NDVI values from -1 to 1)

Figure 6.6: Absolute error maps generated by the prediction models for NDVI (the scale from light to dark represents the absolute error values from 0 to 2)

7 CONCLUSIONS AND FUTURE WORKS

7.1 Conclusions

The application of Sentinel-3 images for Sentinel-2 vegetation products estimation has been assessed by this thesis using machine learning and deep learning framework. To test the feasibility of this work, a paired dataset of 41 S2/S3 same-day images from the year 2019 was prepared and pre-processed to feed into the ML and DL network architectures to predict NDVI and SAVI. Four state of the art ML regression algorithms, linear regression, ridge regression, SVR and Random Forest Regression, and two DL network architectures with different depth and complexities, MLP and Convolutional Neural Network CNN were selected for the study. All the models were implemented and hyperparameter optimization was performed for DL models to find the configuration that works the best for each. Similarly, experimentation was done to improve the implemented CNN architecture [45] by simplifying one of the layers. An analysis was also done to observe the differences in the models caused by the input format of pixel-wise or patch-wise.

The performance of all the models was evaluated using quantitative and qualitative approaches. For a quantitative approach, three metrics namely RMSE, MSE ,and MAE were used as measures of evaluation. Additionally, visual inspection of predicted vegetation product (NDVI) maps were used as a means for qualitative evaluation. The results demonstrate that all the DL architectures except pixel-wise MLP outperformed the ML models with the 3D CNN being the best. Among the ML model, patch-wise RFR and SVR were the best performing ones. MLP with a simple architecture as well as ML linear regression could not perform well which shows that this kind of regression problem requires more complex architectures and algorithms that can map the non-linear relationship well. The experiments performed by varying the patch sizes of samples showed that for the experiments with mid-size resolution imagery such as S3, smaller patch size is preferred. Also, providing patch-wise input to the ML models, as opposed to pixel-wise, improved the performance for the models but with an added computational cost.

In conclusion, this research work assessed the feasibility of using S3 imagery to fill the data gaps in S2 vegetation products using an inter-sensor approach and successfully implemented ML and DL models to fulfill the objectives. The findings from this study are

consistent with literature as mentioned in results and discussion chapter 6. The framework presented is general and with a few changes can be extended for the study of other vegetation products. The models and the prepared dataset can be used for further research that focuses on capitalizing the free and open availability of Sentinel-2 and Sentinel-3 imagery as well as new and advanced technologies to provide better vegetation monitoring capabilities for our planet.

7.2 Future Works

This thesis although limited by time and resource availability, has opened the possibility for future research in inter-sensor data products estimation. Following are the possible ways to improve the results and recommended works for future:

- The framework presented in the study can be extended for the prediction of other products such as biophysical parameters or the inputs and labels can be inversed for prediction of S3 from S2.
- Cloud cover is always a limitation when working with remote sensing imagery over a period of time. Considering data fusion techniques can help to fill the data gaps caused by cloud cover and increase the number of samples to train the model.
- Since CNN structures work well in this scenario, deeper CNN architectures and strategies such as Residual network and Dense network can be considered.
- This study only considers the temporal series of S3. Further research can be done to include the temporal series of both S2 and S3 as an additional dimension in a 3D-CNN architecture.

Bibliography

[1] Z. MALENOVSKỲ, H. ROTT, J. CIHLAR, M. E. SCHAEPMAN, G. GARCÍA-SANTOS, R. FERNANDES, and M. BERGER. "Sentinels for science: Potential of Sentinel-1,-2, and-3 missions for scientific observations of ocean, cryosphere, and land". In: *Remote Sensing of environment* 120 (2012), pp. 91–101.

[2] J. ASCHBACHER and M. P. MILAGRO-PÉREZ. "The European Earth monitoring (GMES) programme: Status and perspectives". In: *Remote Sensing of Environment* 120 (2012), pp. 3–8.

[3] P. KNOEFEL, G. SURESH, J. KRETZ, and M. HOVENBITZER. "Supporting the calculation of SDG indicators using GEO and EO data at the German Federal Agency of Cartography and Geodesy." In: *Geophysical Research Abstracts*. Vol. 21. 2019.

[4] P. DEFOURNY, S. BONTEMPS, N. BELLEMANS, C. CARA, G. DEDIEU, E. GUZZONATO, O. HAGOLLE, J. INGLADA, L. NICOLA, T. RABAUTE, et al. "Near real-time agriculture monitoring at national scale at parcel resolution: Performance assessment of the Sen2-Agri automated system in various cropping systems around the world". In: *Remote sensing of environment* 221 (2019), pp. 551–568.

[5] P. MONDAL, S. S. MCDERMID, and A. QADIR. "A reporting framework for Sustainable Development Goal 15: Multi-scale monitoring of forest degradation using MODIS, Landsat and Sentinel data". In: *Remote Sensing of Environment* 237 (2020), p. 111592.

[6] K. VON SCHUCKMANN, P.-Y. LE TRAON, E. ALVAREZ-FANJUL, L. AXELL, M. BALMASEDA, L.-A. BREIVIK, R. J. BREWIN, C. BRICAUD, M. DREVILLON, Y. DRILLET, et al. "The copernicus marine environment monitoring service ocean state report". In: *Journal of Operational Oceanography* 9.sup2 (2016), s235–s320.

[7] M. DRUSCH, U. DEL BELLO, S. CARLIER, O. COLIN, V. FERNANDEZ, F. GASCON, B. HOERSCH, C. ISOLA, P. LABERINTI, P. MARTIMORT, et al. "Sentinel-2: ESA's optical high-resolution mission for GMES operational services". In: *Remote sensing of Environment* 120 (2012), pp. 25–36.

[8] C. DONLON, B BERRUTI, A BUONGIORNO, M.-H. FERREIRA, P FÉMÉNIAS, J FRERICK, P GORYL, U KLEIN, H LAUR, C MAVROCORDATOS, et al. "The global

monitoring for environment and security (GMES) sentinel-3 mission". In: *Remote Sensing of Environment* 120 (2012), pp. 37–57.

[9] J. MORENO, L. GUANTER, L. ALONSO, L. GOMEZ, J. AMOROS, G. CAMPS, and J. DELEGIDO. "Land science with Sentinel-2 and Sentinel-3 data series synergy". In: *EGUGA* (2010), p. 6805.

[10] D. SCHEFFLER, D. FRANTZ, and K. SEGL. "Spectral harmonization and red edge prediction of Landsat-8 to Sentinel-2 using land cover optimized multivariate regressors". In: *Remote Sensing of Environment* 241 (2020), p. 111723.

[11] R. FERNANDEZ-BELTRAN, F. PLA, and A. PLAZA. "Sentinel-2 and sentinel-3 intersensor vegetation estimation via constrained topic modeling". In: *IEEE Geoscience and Remote Sensing Letters* 16.10 (2019), pp. 1531–1535.

[12] J. M. HAUT, R. FERNANDEZ-BELTRAN, M. E. PAOLETTI, J. PLAZA, A. PLAZA, and F. PLA. "Inter-Sensor Regression Analysis for Operational Sentinel-2 and Sentinel-3 Data Products". In: *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2018, pp. 2603–2606.

[13] X. X. ZHU, D. TUIA, L. MOU, G.-S. XIA, L. ZHANG, F. XU, and F. FRAUNDORFER. "Deep learning in remote sensing: A comprehensive review and list of resources". In: *IEEE Geoscience and Remote Sensing Magazine* 5.4 (2017), pp. 8–36.

[14] B. YU, L. XU, J.-h. PENG, Z. HU, and A. WONG. "Global chlorophyll-a concentration estimation from moderate resolution imaging spectroradiometer using convolutional neural networks". In: *Journal of Applied Remote Sensing* 14.3 (2020), p. 034520.

[15] G. SCARPA, M. GARGIULO, A. MAZZA, and R. GAETANO. "A CNN-based fusion method for feature extraction from sentinel data". In: *Remote Sensing* 10.2 (2018), p. 236.

[16] H. RUSSELLO. "Convolutional neural networks for crop yield prediction using satellite images". In: *IBM Center for Advanced Studies* (2018).

[17] J. VERRELST, J. MUÑOZ, L. ALONSO, J. DELEGIDO, J. P. RIVERA, G. CAMPS-VALLS, and J. MORENO. "Machine learning regression algorithms for biophysical parameter retrieval: Opportunities for Sentinel-2 and-3". In: *Remote Sensing of Environment* 118 (2012), pp. 127–139.

[18] Z. AO, Y. SUN, and Q. XIN. "Constructing 10-m NDVI Time Series From Landsat 8 and Sentinel 2 Images Using Convolutional Neural Networks". In: *IEEE Geoscience and Remote Sensing Letters* (2020).

[19] J. SEGARRA, M. L. BUCHAILLOT, J. L. ARAUS, and S. C. KEFAUVER. "Remote sensing for precision agriculture: Sentinel-2 improved features and applications". In: *Agronomy* 10.5 (2020), p. 641.

[20] H. K. LICHTENTHALER, M LANG, M SOWINSKA, F HEISEL, and J. MIEHE. "Detection of vegetation stress via a new high resolution fluorescence imaging system". In: *Journal of plant physiology* 148.5 (1996), pp. 599–612.

[21] A. A. GITELSON, A. VIÑA, S. B. VERMA, D. C. RUNDQUIST, T. J. ARKEBAUER, G. KEYDAN, B. LEAVITT, V. CIGANDA, G. G. BURBA, and A. E. SUYKER. "Relationship between gross primary production and chlorophyll content in crops: Implications for the synoptic monitoring of vegetation productivity". In: *Journal of Geophysical Research: Atmospheres* 111.D8 (2006).

[22] J. XUE and B. SU. "Significant remote sensing vegetation indices: A review of developments and applications". In: *Journal of sensors* 2017 (2017).

[23] R. S. LUNETTA, J. F. KNIGHT, J. EDIRIWICKREMA, J. G. LYON, and L. D. WORTHY. "Land-cover change detection using multi-temporal MODIS NDVI data". In: *Remote sensing of environment* 105.2 (2006), pp. 142–154.

[24] L. P. VENANCIO, E. C. MANTOVANI, C. H. do AMARAL, C. M. U. NEALE, I. Z. GONÇALVES, R. FILGUEIRAS, and I. CAMPOS. "Forecasting corn yield at the farm level in Brazil based on the FAO-66 approach and soil-adjusted vegetation index (SAVI)". In: *Agricultural Water Management* 225 (2019), p. 105779.

[25] W. J. FRAMPTON, J. DASH, G. WATMOUGH, and E. J. MILTON. "Evaluating the capabilities of Sentinel-2 for quantitative estimation of biophysical variables in vegetation". In: *ISPRS journal of photogrammetry and remote sensing* 82 (2013), pp. 83–92.

[26] M. LANGE, B. DECHANT, C. REBMANN, M. VOHLAND, M. CUNTZ, and D. DOKTOR. "Validating MODIS and sentinel-2 NDVI products at a temperate deciduous forest site using two independent ground-based sensors". In: *Sensors* 17.8 (2017), p. 1855.

[27] K. A. AL-GAADI, A. A. HASSABALLA, E. TOLA, A. G. KAYAD, R. MADUGUNDU, B. ALBLEWI, and F. ASSIRI. "Prediction of potato crop yield using precision agriculture techniques". In: *PloS one* 11.9 (2016), e0162219.

[28] R. A. PURNAMASARI, R. NOGUCHI, and T. AHAMED. "Land suitability assessments for yield prediction of cassava using geospatial fuzzy expert systems and remote sensing". In: *Computers and Electronics in Agriculture* 166 (2019), p. 105018.

[29] Q. WANG and P. M. ATKINSON. "Spatio-temporal fusion for daily Sentinel-2 images". In: *Remote Sensing of Environment* 204 (2018), pp. 31–42.

[30] I. CAZZANIGA, M. BRESCIANI, R. COLOMBO, V. DELLA BELLA, R. PADULA, and C. GIARDINO. "A comparison of Sentinel-3-OLCI and Sentinel-2-MSI-derived Chlorophyll-a maps for two large Italian lakes". In: *Remote Sensing Letters* 10.10 (2019), pp. 978–987.

[31] C. HUANG, L. DAVIS, and J. TOWNSHEND. "An assessment of support vector machines for land cover classification". In: *International Journal of remote sensing* 23.4 (2002), pp. 725–749.

[32] A. CHLINGARYAN, S. SUKKARIEH, and B. WHELAN. "Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review". In: *Computers and electronics in agriculture* 151 (2018), pp. 61–69.

[33] J. VERRELST, J. P. RIVERA, F. VEROUSTRAETE, J. MUÑOZ-MARÍ, J. G. CLEVERS, G. CAMPS-VALLS, and J. MORENO. "Experimental Sentinel-2 LAI estimation using parametric, non-parametric and physical retrieval methods–A comparison". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 108 (2015), pp. 260–272.

[34] O. KIRA, A. L. NGUY-ROBERTSON, T. J. ARKEBAUER, R. LINKER, and A. A. GITELSON. "Toward generic models for green LAI estimation in maize and soybean: Satellite observations". In: *Remote Sensing* 9.4 (2017), p. 318.

[35] D. UPRETI, W. HUANG, W. KONG, S. PASCUCCI, S. PIGNATTI, X. ZHOU, H. YE, and R. CASA. "A comparison of hybrid machine learning algorithms for the retrieval of wheat biophysical variables from sentinel-2". In: *Remote Sensing* 11.5 (2019), p. 481.

[36] K. HE, X. ZHANG, S. REN, and J. SUN. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[37] S. LATHUILIÈRE, P. MESEJO, X. ALAMEDA-PINEDA, and R. HORAUD. "A comprehensive analysis of deep regression". In: *IEEE transactions on pattern analysis and machine intelligence* 42.9 (2019), pp. 2065–2081.

[38] Y. LIU, X. CHEN, Z. WANG, Z. J. WANG, R. K. WARD, and X. WANG. "Deep learning for pixel-level image fusion: Recent advances and future prospects". In: *Information Fusion* 42 (2018), pp. 158–173.

[39] T. KATTENBORN, J. LEITLOFF, F. SCHIEFER, and S. HINZ. "Review on Convolutional Neural Networks (CNN) in vegetation remote sensing". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 173 (2021), pp. 24–49.

[40] O. E. APOLO-APOLO, M. PÉREZ-RUIZ, J. MARTÍNEZ-GUANTER, and G. EGEA. "A mixed data-based deep neural network to estimate leaf area index in wheat breeding trials". In: *Agronomy* 10.2 (2020), p. 175.

[41] D. TRAN, L. BOURDEV, R. FERGUS, L. TORRESANI, and M. PALURI. "Learning spatiotemporal features with 3d convolutional networks". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 4489–4497.

[42] S. JI, C. ZHANG, A. XU, Y. SHI, and Y. DUAN. "3D convolutional neural networks for crop classification with multi-temporal remote sensing images". In: *Remote Sensing* 10.1 (2018), p. 75.

[43] M. PAOLETTI, J. HAUT, J PLAZA, and A PLAZA. "Deep learning classifiers for hyperspectral imaging: A review". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 158 (2019), pp. 279–317.

[44] Y. LECUN, L. BOTTOU, Y. BENGIO, and P. HAFFNER. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[45] A. SHARMA, X. LIU, X. YANG, and D. SHI. "A patch-based convolutional neural network for remote sensing image classification". In: *Neural Networks* 95 (2017), pp. 19–28.

[46] C KALAITZIDIS, V HEINZEL, and D ZIANIS. "A review of multispectral vegetation indices for biomass estimation". In: *Proceedings of the 29th symposium of the European association of remote sensing laboratories, Chania, Greece. IOS Press Ebook*. 2010, pp. 201–208.

[47] J. ROUSE, R. H. HAAS, J. A. SCHELL, D. W. DEERING, et al. "Monitoring vegetation systems in the Great Plains with ERTS". In: *NASA special publication* 351.1974 (1974), p. 309.

[48] A. R. HUETE. "A soil-adjusted vegetation index (SAVI)". In: *Remote sensing of environment* 25.3 (1988), pp. 295–309.

[49] P. RHYMA, K NORIZAH, O HAMDAN, I FARIDAH-HANUM, and A. ZULFA. "Integration of normalised different vegetation index and soil-adjusted vegetation index for mangrove vegetation delineation". In: *Remote Sensing Applications: Society and Environment* 17 (2020), p. 100280.

[50] SENTINEL-HUB. *Normalized difference vegetation index | Sentinel-Hub custom scripts*. https://custom-scripts.sentinel-hub.com/custom-scripts/sentinel-2/ndvi/.

[51] SENTINEL-HUB. *SAVI (Soil Adjusted Vegetation Index) | Sentinel-Hub custom scripts*. https://custom-scripts.sentinel-hub.com/custom-scripts/sentinel-2/savi/.

[52] D. J. LARY, A. H. ALAVI, A. H. GANDOMI, and A. L. WALKER. "Machine learning in geosciences and remote sensing". In: *Geoscience Frontiers* 7.1 (2016), pp. 3–10.

[53] A. E. HOERL and R. W. KENNARD. "Ridge regression: Biased estimation for nonorthogonal problems". In: *Technometrics* 12.1 (1970), pp. 55–67.

[54] H. DRUCKER, C. J. BURGES, L. KAUFMAN, A. SMOLA, V. VAPNIK, et al. "Support vector regression machines". In: *Advances in neural information processing systems* 9 (1997), pp. 155–161.

[55] A. J. SMOLA and B. SCHÖLKOPF. "A tutorial on support vector regression". In: *Statistics and computing* 14.3 (2004), pp. 199–222.

[56] L. BREIMAN. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[57] V. RODRIGUEZ-GALIANO, M. P. MENDES, M. J. GARCIA-SOLDADO, M. CHICA-OLMO, and L. RIBEIRO. "Predictive modeling of groundwater nitrate pollution using Random Forest and multisource variables related to intrinsic and specific vulnerability: A case study in an agricultural setting (Southern Spain)". In: *Science of the Total Environment* 476 (2014), pp. 189–206.

[58] J. SCHMIDHUBER. "Deep learning in neural networks: An overview". In: *Neural networks* 61 (2015), pp. 85–117.

[59] K. O'SHEA and R. NASH. "An introduction to convolutional neural networks". In: *arXiv preprint arXiv:1511.08458* (2015).

[60] G. GREKOUSIS. "Artificial neural networks and deep learning in urban geography: A systematic review and meta-analysis". In: *Computers, Environment and Urban Systems* 74 (2019), pp. 244–256.

[61] F. MURTAGH. "Multilayer perceptrons for classification and regression". In: *Neurocomputing* 2.5-6 (1991), pp. 183–197.

[62] K. FUKUSHIMA and S. MIYAKE. "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition". In: *Competition and cooperation in neural nets*. Springer, 1982, pp. 267–285.

[63] V. NAIR and G. E. HINTON. "Rectified linear units improve restricted boltzmann machines". In: *Icml*. 2010.

[64] D. P. KINGMA and J. BA. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[65] F. J. JARAÍZ-CABANILLAS, J. MORA-ALISEDA, J. S. JEONG, and J. GARRIDO-VELARDE. "Methodological proposal to classify and delineate natural protected areas. Study case: Region of Extremadura, Spain". In: *Land Use Policy* 79 (2018), pp. 310–319.

[66] J.-M. SÁNCHEZ-MARTÍN, R. BLAS-MORATO, and J.-I. RENGIFO-GALLEGO. "The dehesas of extremadura, Spain: A potential for socio-economic development based on agritourism activities". In: *Forests* 10.8 (2019), p. 620.

[67] ESA. *Sentinel-2 User Handbook, Issue 1, Rev 2*. Tech. rep. ESA Standard Document. European Space Agency, 2015.

[68] ESA. *Copernicus open access hub*. https://scihub.copernicus.eu/dhus.

[69] D. R. PEDDLE, H. P. WHITE, R. J. SOFFER, J. R. MILLER, and E. F. LEDREW. "Reflectance processing of remote sensing spectroradiometer data". In: *Computers & geosciences* 27.2 (2001), pp. 203–213.

[70] B. C. GMBH. *2. The SNAP S3-SNOW Processors — SEOM S3 for Snow / SICE*. https://s3tbx-snow.readthedocs.io/en/latest/s3snow_processing_system.html.

[71] T. TAKASE, S. OYAMA, and M. KURIHARA. “Why does large batch training result in poor generalization? A comprehensive explanation and a better strategy from the viewpoint of stochastic optimization”. In: *Neural computation* 30.7 (2018), pp. 2005–2023.

[72] H. SONG, Y. KIM, and Y. KIM. “A patch-based light convolutional neural network for land-cover mapping using Landsat-8 images”. In: *Remote Sensing* 11.2 (2019), p. 114.

Masters Program in Geospatial Technologies

ENHANCING TEMPORAL SERIES OF SENTINEL-2 AND SENTINEL-3 DATA PRODUCTS: FROM CLASSICAL REGRESSION TO DEEP LEARNING APPROACH

Anu Bhalu Shrestha

Dissertation submitted in partial fulfilment of the requirements
for the Degree of *Master of Science in Geospatial Technologies*

NOVA IMS Information Management School
Universitat Jaume I
Westfälische Wilhelms-Universität Münster

Masters Program in **Geospatial Technologies**

Supported by:

ERASMUS MUNDUS