sid.inpe.br/mtc-m21c/2021/05.05.01.16-TDI

LAND USE AND LAND COVER CLASSIFICATION OF SATELLITE IMAGE TIME SERIES USING MACHINE LEARNING

Rolf Ezequiel de Oliveira Simões

Doctorate Thesis of the Graduate Course in Applied Computing, guided by Drs. Gilberto Camara Neto, and Gilberto Ribeiro de Queiroz, approved in May 12, 2021.

URL of the original document:
<http://urlib.net/8JMKD3MGP3W34R/44KLMUS>

INPE
São José dos Campos
2021

PUBLISHED BY:

Instituto Nacional de Pesquisas Espaciais - INPE
Coordenação de Ensino, Pesquisa e Extensão (COEPE)
Divisão de Biblioteca (DIBIB)
CEP 12.227-010
São José dos Campos - SP - Brasil
Tel.:(012) 3208-6923/7348
E-mail: pubtc@inpe.br

BOARD OF PUBLISHING AND PRESERVATION OF INPE INTELLECTUAL PRODUCTION - CEPPII (PORTARIA Nº 176/2018/SEI-INPE):

Chairperson:

Dra. Marley Cavalcante de Lima Moscati - Coordenação-Geral de Ciências da Terra (CGCT)

Members:

Dra. Ieda Del Arco Sanches - Conselho de Pós-Graduação (CPG)

Dr. Evandro Marconi Rocco - Coordenação-Geral de Engenharia, Tecnologia e Ciência Espaciais (CGCE)

Dr. Rafael Duarte Coelho dos Santos - Coordenação-Geral de Infraestrutura e Pesquisas Aplicadas (CGIP)

Simone Angélica Del Ducca Barbedo - Divisão de Biblioteca (DIBIB)

DIGITAL LIBRARY:

Dr. Gerald Jean Francis Banon

Clayton Martins Pereira - Divisão de Biblioteca (DIBIB)

DOCUMENT REVIEW:

Simone Angélica Del Ducca Barbedo - Divisão de Biblioteca (DIBIB)

André Luis Dias Fernandes - Divisão de Biblioteca (DIBIB)

ELECTRONIC EDITING:

Ivone Martins - Divisão de Biblioteca (DIBIB)

André Luis Dias Fernandes - Divisão de Biblioteca (DIBIB)

sid.inpe.br/mtc-m21c/2021/05.05.01.16-TDI

LAND USE AND LAND COVER CLASSIFICATION OF SATELLITE IMAGE TIME SERIES USING MACHINE LEARNING

Rolf Ezequiel de Oliveira Simões

Doctorate Thesis of the Graduate Course in Applied Computing, guided by Drs. Gilberto Camara Neto, and Gilberto Ribeiro de Queiroz, approved in May 12, 2021.

URL of the original document:
<http://urlib.net/8JMKD3MGP3W34R/44KLMUS>

INPE
São José dos Campos
2021

Cataloging in Publication Data

Simões, Rolf Ezequiel de Oliveira.

Si51l Land use and land cover classification of satellite image time series using machine learning / Rolf Ezequiel de Oliveira Simões. – São José dos Campos : INPE, 2021.

xxii + 67 p. ; (sid.inpe.br/mtc-m21c/2021/05.05.01.16-TDI)

Thesis (Doctorate in Applied Computing) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2021.

Guiding : Drs. Gilberto Camara Neto, and Gilberto Ribeiro de Queiroz.

1. Big Earth observation data. 2. Satellite image time series. 3. Machine learning. 4. Land use and land cover maps. 5. Cloud computing. I.Title.

CDU 004.032.2:528.8

Esta obra foi licenciada sob uma Licença Creative Commons Atribuição-NãoComercial 3.0 Não Adaptada.

This work is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported License.

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

Serviço de Pós-Graduação - SEPGR

DEFESA FINAL DE TESE DE ROLF EZEQUIEL DE OLIVEIRA SIMÕES
BANCA Nº 051/2021, REG 136778/16

No dia 12 de maio de 2021, as 10 horas, por teleconferência, o(a) aluno(a) mencionado(a) acima defendeu seu trabalho final (apresentação oral seguida de arguição) perante uma Banca Examinadora, cujos membros estão listados abaixo. O(A) aluno(a) foi APROVADO(A) pela Banca Examinadora, por unanimidade, em cumprimento ao requisito exigido para obtenção do Título de Doutor em Computação Aplicada. O trabalho precisa da incorporação das correções sugeridas pela Banca Examinadora e revisão final pelo(s) orientador(es).

Título: "Land use and land cover classification of satellite image time series using machine learning."

Eu, Rafael Duarte Coelho dos Santos, como Presidente da Banca Examinadora, assino esta ATA em nome de todos os membros, com o consentimento dos mesmos.

Membros da banca:

Dr. Rafael Duarte Coelho dos Santos - Presidente - INPE
Dr. Gilberto Camara Neto - Orientador – INPE
Dr. Gilberto Ribeiro de Queiroz - Orientador – INPE
Dr. Claudio Aparecido Almeida - Membro Interno – INPE
Dr. Damien Arvor - Membro Externo – CNRS
Dra. Ana Carolina Lorena - Membro Externo - ITA

Documento assinado eletronicamente por **Rafael Duarte Coelho dos Santos**, **Tecnologista**, em 14/05/2021, às 15:14 (horário oficial de Brasília), com fundamento no art. 6º do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.mctic.gov.br/verifica.html, informando o código verificador **6857957** e o código CRC **9912E4DC**.

Referência: Processo nº 01340.001864/2021-93 SEI nº 6857957

"...assim é que a vida deve ser, quando um desanima, o outro agarra-se às próprias tripas e faz delas coração".

José Saramago
in *"A Caverna"*

For my parents Sandra and Inácio, my wife Cristiane, and my children Sarah, Raquel, and Saulo

ACKNOWLEDGEMENTS

Firstly, I would like to thank my fellow lab mates at INPE, Michelle Picoli for showing me the importance of getting things done the way I could and not in a perfect way, Alber Sanchez for spending a lot of his time debugging code with me and presenting me nice dev tools that made my work easier. Michel Chaves for the friendship and insightful talks, Rodrigo Begotti for the long conversations and laughs that make my hard days better. Lorena Santos for the long partnership and company since the starting of my PhD. Felipe Carvalho for the partnership and enthusiasm for new projects. For all of them, I am very thankful for the good times and work collaborations.

I am very grateful to my INPE colleagues and friends Mikhaela Plestch, Henrique Rennó, Jéssica Domingues, Carlos Romani, Guilherme Chagas, Renan Marujo, and Jano Simas for spent their time with me on long and good conversations and laughing that made my days at INPE even better.

I could not forget to express my sincere gratitude for the welcome I received and the moments shared with Moshé Cotacallapa and Carolline Tressmann at our student house, the "República Saint Moritz 74". Those were great and unforgettable days.

I would like to express my sincere and deep gratitude for my research supervisors Dr. Gilberto Camara and Dr. Gilberto Queiroz, for their invaluable guidance in my intensive journey in INPE and giving me opportunities to do my research with all the support I needed. It was a great privilege to work and study with them. I would also like to thank their friendship and the delightful social moments spent with them. I extend my cordial thanks to their wives for sharing these moments. I would also like to say that Dr. Gilberto Camara has deeply inspired me with his dynamism, vision, scientific knowledge, and appreciation for the arts. I am very grateful for his partnership, patience, and dedication to discuss this research work and thesis preparation.

Finally, I would like to thanks the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) for the financial support (grant 140684/2016-6).

ABSTRACT

Human activities are impacting the global environment and changing the landscape of the Earth at an unprecedented pace. One way to understand and quantify the impacts of these transformations is to measure land cover and land use changes. Earth observation by orbital sensors has became the most consistent way to do this task as it reaches all globe in a periodic manner. The planet is continuously monitored and large image databases are open to the public community. Cloud computing is becoming the main choice for storage, distribution, and processing of satellite imagery. To use these data sets to understand how our planet is changing, researchers are developing software for big Earth observation data analytics. The massive volume of spatiotemporal digital assets has brought about a major computational challenge: *how to design and build technologies that allow the Earth observation community to analyse big data sets?* This thesis addresses this problem by proposing an open source *R* package called *sits* . The package works with satellite image time series and machine learning methods to produce land use and land cover classification in a "time-first, space-later" approach. Our hypothesis is that using all observed spectral values of the time series coupled with advanced statistical learning methods is a robust approach for land classification. Two case studies show that this approach produces results of high accuracy. The first was conducted in Mato Grosso State, Brazil, for years 2001-2017. The second was conducted in Cerrado biome for the year 2018. These study areas comprise some of the world's most dynamic agricultural frontiers. To show how to access collections of satellite images, we also present the *rstac R* package. This package allows *sits* to connect to different imagery repositories on the cloud.

Keywords: Big Earth observation data. Satellite image time series. Machine learning. Land use and land cover maps. Cloud computing.

CLASSIFICAÇÃO DE USO E COBERTURA DA TERRA A PARTIR DE SÉRIES TEMPORAIS DE IMAGENS DE SATÉLITE USANDO MÉTODOS DE APRENDIZADO DE MÁQUINA

RESUMO

As atividades humanas estão impactando o meio ambiente global e mudando a paisagem da Terra a um ritmo sem precedentes. Uma maneira de entender e quantificar os impactos destes transformações é olhar para a cobertura e as mudanças de uso do solo. Nas últimas décadas, imagens de satélite sensores orbitais tornaram-se a maneira mais consistente de faze-lo. Atualmente, o planeta é continuamente monitorado e várias bases de dados de imagens estão abertas à comunidade científica. Para armazenar, distribuição e processar essas imagens, a principal escolha é o uso de computação em nuvem. Esse enorme volume de dados trouxe um desafio computacional: *como projetar e construir tecnologias que permitam à comunidade de observação da Terra analisar grandes conjuntos de dados?* Esta tese contribui para resolver esse desafio científico ao propor um pacote *R* de código aberto chamado *sits* . O software usa séries temporais de imagens de satélite e métodos de aprendizagem de máquinas para produzir a classificação de uso e cobertura da Terra em uma abordagem "time-first, space-later". Nossa hipótese é a de que usar séries temporais juntamente com métodos avançados de aprendizagem de máquina é uma abordagem robusta para a classificação do uso e cobertura da terra. Mostramos no trabalho que esta abordagem produz mapas de uso e cobertura da Terra com boa precisão através de dois estudos de caso. O primeiro foi realizado no estado do Mato Grosso, Brasil, para o período 2001-2017. O segundo no bioma brasileiro do Cerrado para o ano de 2018. Nessas áreas de estudo estão localizas algumas das fronteiras agrícolas mais dinâmicas do mundo. Além disso, apresentamos o pacote *rstac* desenvolvido e integrado no *sits* que nos permitiu conectar a diferentes repositórios de imagens na nuvem.

Keywords: Big data. Observação da Terra. Séries temporais de imagens de satélites. Aprendizado de máquina. Mapas de uso e cobertura da terra. Computação em nuvem.

LIST OF FIGURES

Page

LIST OF TABLES

Page

LIST OF ABBREVIATIONS

API	–	Application programming interface
ARD	–	Analysis ready data
AWS	–	Amazon Web Service
BDC	–	Brazil Data Cube
CRAN	–	The comprehensive R archive network
EVI	–	Enhanced vegetation index
GHG	–	Greenhouse gas emissions
IBGE	–	Brazilian Institute of Geography and Statistics
INPE	–	National Institute for Space Research
LULC	–	Land use and land cover
MGRS	–	Military grid reference system
MIR	–	Mid-infrared
ML	–	Machine learning
NDC	–	Nationally Determined Contribution
NDVI	–	Normalised difference vegetation index
NIR	–	Near-infrared
OAFeat	–	OGC API Features core specification
ODC	–	Open Data Cube
OGC	–	Open Geospatial Consortium
PRODES	–	Deforestation monitoring project
RBF	–	Radial basis kernel function
REST	–	Representational state transfer software architecture
RESTful	–	A Web service that follows the guidelines of REST
RNN	–	Recurrent neural networks
SITS	–	Satellite image time series
STAC	–	Spatio-temporal asset catalogs
SVM	–	Support vector machine
TempCNN	–	Temporal convolutional neural network

CONTENTS

Page

1 INTRODUCTION

Changes in land have a great impact on carbon dioxide emissions (STOCKER et al., 2013; ARNETH et al., 2017), biodiversity loss (FLYNN et al., 2009), ecosystems degradation (FOLEY et al., 2005), and water availability and quality (SCANLON et al., 2007). Land use and land cover change studies are a central field of investigation in the Sustainability Science. These studies provide clear indications on human activities and its impact on landscapes and natural resources of the planet. Thus, land cover products provide support for environmental monitoring and public policy (WULDER et al., 2020).

The primary source of information about land use and land cover changes comes from Earth observation data. Most space agencies have adopted an open data policy, making available an unprecedented amount of satellite imagery to the public (WULDER et al., 2012). Combined with advances in machine learning and cloud computing, experts can now process large volumes of earth observation data (PARENTE et al., 2019; AMANI et al., 2020; FERREIRA et al., 2020a; GOMES et al., 2020). These massive data sets allow better understanding of global changes, but bring about new challenges. Researchers are calling for new big data analytics (CAMARA et al., 2016) focusing particularly on satellite image time series (KUENZER et al., 2015). This thesis addresses the problem of producing reliable land cover products from satellite image time series. Thus, our scientific question is: *how to design and build technologies that allow the Earth observation community to analyse big data sets?*

Our answer to that scientific question is to design and develop an open source *R* package called *sits* that works with satellite image time series and machine learning methods to produce land use and land cover classification in a "time-first, space-later" approach. All observations are used to derive land use and land cover maps. Our hypothesis is that using all observed spectral values of the time series coupled with advanced statistical learning methods is a robust approach for land use and land cover classification of satellite image time series. The software is open and efficient and not locked to specific hardware architecture. It runs in any cloud computing environment, taking advantage of existing Earth observation data collections.

Developers of analytical software for big Earth observation data face several challenges. Designers need to balance between conflicting factors. Solutions which are efficient for specific hardware architectures can not be used in other environments. Packages based on generic hardware and open standards will not have the same performance as dedicated solutions. Software that assumes that its users are computer programmers

are flexible, but may be difficult to learn for a wide audience. When designing and implementing *sits* , we aimed to reach a large audience of experts, by providing a compact and expressive API that runs on multiple environments.

Three papers compose this thesis. The first Chapter is a manuscript submitted to the special issue "Big Earth Observation Data: From Cloud Technologies to Insights and Foresight" of the journal Remote Sensing journal. This paper describes *sits* , its software design, API workflow, and how it does data access and parallel processing in cloud computing environments. It shows how *sits* provides support for the classification method using satellite image time series. We present a case study in Brazilian Cerrado biome, producing a land use and land cover map for year 2018. We also discuss the trade-offs of our choices on *sits* software design compared to other Earth observation known solutions.

The second Chapter is a paper published in the journal Nature Scientific Data (SIMOES et al., 2020). It presents a case study of land use and land cover classification from 2001 to 2017 in the state of Mato Grosso, Brazil. It describes methods to obtain land cover products of high accuracy using time series and machine learning. We published the data and results in an open repository. Results provide qualified information to researchers on crop and pasture expansion over natural vegetation, on agricultural productivity, and on interplay between production and protection.

The third Chapter is a version of a paper accepted by the 2021 International Geoscience and Remote Sensing Symposium (IGARSS 2021). It describes *rstac* , an *R* package that implements a client for the SpatioTemporal Assets Catalog (STAC) protocol. This protocol describes geospatial information available in big image collections (HANSON, 2019). The *rstac* package is an open RESTful API web service based on OGC OAFeat that enables *sits* to connect to different image cloud repositories. We discuss how common protocols are critical to interoperability in heterogeneous cloud computing environments. We show how *rstac* enables *sits* to query and find all images needed to perform a land use and land cover classification.

2 SATELLITE IMAGE TIME SERIES ANALYSIS FOR BIG EARTH OBSERVATION DATA

This Chapter[1] describes *sits* , an open-source *R* package for satellite image time series analysis using machine learning. To allow experts to use the full extent of satellite imagery, *sits* adopts a *time-first, space-later* approach. It supports the complete cycle of data analysis for land classification. Its API provides a simple but powerful set of functions. The software works in different cloud computing environments. Satellite image time series are input to machine learning classifiers, and the results are post-processed using spatial smoothing. Since machine learning methods need accurate training data, *sits* includes methods for quality assessment of training samples. The software also provides methods for validation and accuracy measurement. The package thus comprises a production environment for big Earth Observation data analysis. We show that this approach produces high accuracy for land use and land cover maps by a case study in Cerrado biome, one of the world's fast moving agricultural frontiers, for the year 2018.

2.1 Introduction

The growing demand for natural resources has caused major environmental impacts and is changing landscapes everywhere. Conversion of land cover due to human use is one of the key factors behind greenhouse gas emissions and biodiversity loss (FOLEY et al., 2005). Spatial quantification of land use and land cover change allows societies to understand the extent of these impacts. Satellites are required to generate land cover products, since they provide a consistent, periodic, and globally reaching coverage of the planet's surface. Thus, satellite-based land cover products are essential to support evidence-based policies that promote sustainability.

There is currently an extensive amount of Earth observation (EO) data collected by an increasing number of satellites. Coupled with the adoption of open data policies by most spatial agencies, an unprecedented amount of satellite data is now publicly available (WULDER et al., 2012). This has brought a significant challenge for researchers and developers of geospatial technologies: *how to design and build technologies that allow the Earth observation community to analyse big data sets?*

[1] A version of this Chapter was submitted to the special issue "Big Earth Observation Data: From Cloud Technologies to Insights and Foresight" of the journal Remote Sensing. The paper is authored by Rolf Simoes, Gilberto Camara, Gilberto Queiroz, Felipe Souza, Pedro R. Andrade, Lorena Santos, Alexandre Carvalho and Karine Ferreira.

The emergence of cloud computing services capable of storing and processing big EO data sets allows researchers to develop innovative methods for extracting information (GORELICK et al., 2017; GIULIANI et al., 2019). One of the relevant trends is to work with satellite image time series, which are calibrated and comparable measures of the same location on Earth at different times. These measures can come from a single sensor (e.g., MODIS) or by combining various sensors (e.g., Landsat 8 and Sentinel-2). When associated with frequent revisits, image time series can capture significant land use and land cover changes (VERBESSELT et al., 2010). For this reason, developing methods to analyse image time series has become a relevant research area in Remote Sensing (ARVOR et al., 2011; MAUS et al., 2016; PELLETIER et al., 2019).

Multiyear time series of land cover attributes enable a broader view of land change. Time series capture both gradual and abrupt changes (LAMBIN et al., 2003). Researchers have used time series in applications such as forest disturbance (KENNEDY et al., 2010), land change (ZHU et al., 2020), ecological dynamics (PASQUARELLA et al., 2016), agricultural intensification (GALFORD et al., 2008), and deforestation monitoring (ARVOR et al., 2012).

Methods for processing image time series include BFAST for detecting trends and breaks (VERBESSELT et al., 2010), TIMESAT for extracting phenological attributes (JONSSON; EKLUNDH, 2004), CCDC for continuous change detection (ARÉVALO et al., 2020), and methods based on Dynamic Time Warping (DTW) for land use and land cover classification (PETITJEAN et al., 2012; MAUS et al., 2019). Machine learning methods such as random forests and support vector machines have also been proposed and tested (PELLETIER et al., 2016; SHEEREN et al., 2016). More recently, researchers have proposed deep learning classifiers designed specifically to work with image time series (RUSSWURM; KORNER, 2018; PELLETIER et al., 2019; GARNOT; LANDRIEU, 2020). However, these methods are self-contained. Interested users need an additional effort to access and prepare the input data to use them. Ideally, such methods should be included in an end-to-end environment that provides data handling and management.

This Chapter describes *sits* , an open-source *R* package for satellite image time series analysis using machine learning. It supports the complete cycle of data analysis for land classification, including data management, validation and quality assessment, filtering, classification, post-processing, and accuracy estimates. Its API provides a simple but powerful set of functions.

The traditional approach for change detection in remote sensing is to compare two classified images of the same place in different times and derive a transition matrix. Camara et al. (CAMARA et al., 2016) call this a *space-first, time-later* approach. The *sits* package adopts a *time-first, space-later* method, where all the values of time series are inputs for analysis and thus track changes continuously (SUDMANNS et al., 2018; WOODCOCK et al., 2020). In the *time-first, space-later* method each spatial location is associated to a time series. Each time series is classified separately and spatial smoothing is later applied. The resulting maps of land use and land cover capture the full information provided by big EO data sets.

The *sits* package makes extensive use of the *R* software ecosystem and has been designed for extensibility so that contributors can add new data analysis methods. In this Chapter, we show that algorithms such as TempCNN (PELLETIER et al., 2019) and ResNet (FAWAZ et al., 2019) can be integrated into *sits* . These examples point out how the open and extensible API of *sits* allows researchers to develop advanced methods for big EO data analysis.

We organise this Chapter as follows. In Section 2, we review Earth observation data cubes, pointing out the challenges involved in building them. Section 3 presents the design decisions for the *sits* API and the internal components of the package. Section 4 shows a concrete example of using *sits* to perform land use and land cover classification in the Brazilian Cerrado and discusses the lessons learned. We conclude by pointing out further directions in the development of the package.

2.2 Earth observation data cubes

The term *EO data cube* is being widely used to refer to large collections of satellite images modelled as multidimensional structures to support time series analysis in an easy way to scientists (APPEL; PEBESMA, 2019). There are different definitions of EO data cube. Some authors refer to EO data cubes as organised collections of images (LEWIS et al., 2017) or to the software used to produce the data collection (GIULIANI et al., 2020). Others are more restrictive, defining data cubes as regular collections reprocessed to a common projection and a consistent timeline (FERREIRA et al., 2020a; APPEL; PEBESMA, 2019). We propose a *conceptual* approach, following the idea of EO data cubes as *geographical fields* (GALTON, 2004; CAMARA et al., 2014). The essential property of a geographical field is its field function; for each location within a spatiotemporal extent, this function produces a set of values. This perspective leads to the following definitions:

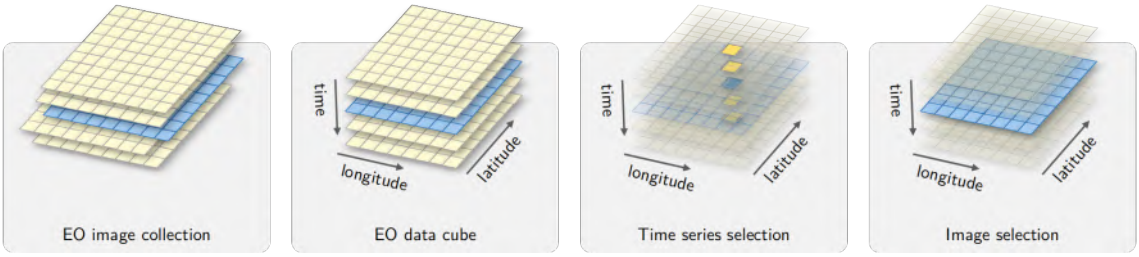
Definition 1. *A data cube is defined by a field function* $f : p \rightarrow \mathbf{v}, \forall p \in ST, \exists\, \mathbf{v}$*, where* ST *is a set of positions in space-time and* $\mathbf{v}$ *is a vector of attributes.*
Definition 2. *An Earth observation data cube DC is a data cube whose spatiotemporal extent has a two-dimensional spatial component* $S : \mathbf{X} \times \mathbf{Y}$ *where* $\forall p = (x_i, y_j) \in S$*, the point* p *can be referenced to a location on the surface of the Earth.*
Definition 3. *The temporal component of the spatiotemporal extent* ST *is a set of time intervals* $\mathbf{T} = t_1, ..., t_n$ *such that* $\forall(i, j, i \neq j)$, $t_i \cap t_j = \emptyset$ *and* $\forall(i, i+1), Meets(t_i, t_{i+1})$*, where* $Meets(.)$ *is the temporal relation defined by Allen and Ferguson (ALLEN; FERGUSON, 1994).*

Definitions [1-3] capture the essential properties of an EO data cube: (a) there is a unique field function; (b) the spatial support is georeferenced; (c) temporal continuity is assured; and (d) all spatiotemporal locations share the same set of attributes (See Figure 2.1). Since the proposed definition is an abstract one, it can be satisfied by different concrete implementations.

Given their properties, data cubes provide a useful abstraction for algorithms that extract information from big EO data sets. Most machine learning and deep learning algorithms require the input data to be consistent. The dimensionality of the data used for training the model has to be the same as that of the data to be classified. There should be no gaps in the input data and no missing values are allowed. However, image collections available in cloud services do not support these requirements. Thus, software developers need to provide an additional abstraction layer. To better understand the problem, consider the differences between analysis ready data (ARD) *image collections* and *data cubes*:
Definition 4. *An ARD image collection is a set of files from a given sensor (or a combined set of sensors) that has been corrected to ensure comparable measurements between different dates. All images are reprojected to a single cartographical projection following well-established standards. Data producers usually crop ARD image collections into tiling systems.*

ARD image collections do not fully support a field function, as required by Definition 1 of data cubes. These collections do not guarantee that every pixel of an image has a valid set of values, since they may contain cloudy and missing pixels. For example, Figure 2.2 shows images of MGRS tile 20LKP of the Sentinel-2/2A image collection available in AWS for different dates. Some images have a significant number of clouds. To support the data cube abstraction, data analysis software has to replace cloudy or missing pixels by valid values.

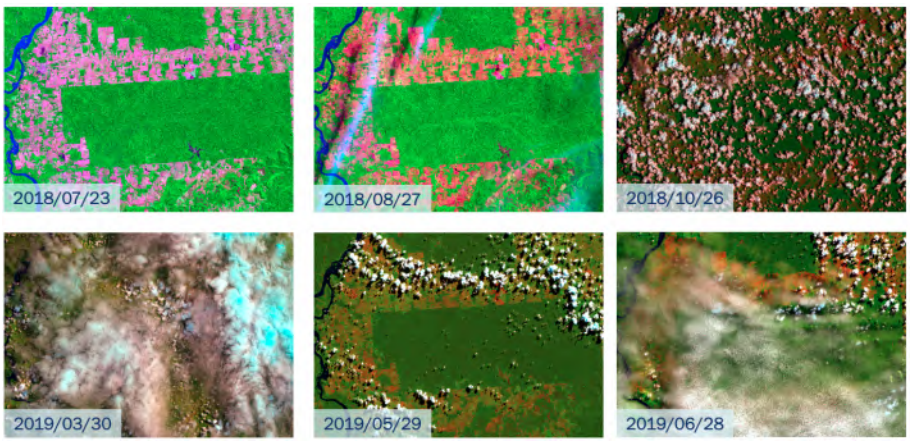
Figure 2.1 - Conceptual view of data cubes.



A further point concerns the timeline of different tiles. Consider the neighbouring Sentinel-2 tiles 20LLP and 20LKP for the period 2018-07-13 to 2019-07-28. Tile 20LLP has 144 temporal instances, while tile 20LKP has only 71 instances. Such differences in temporal extent are common in large image collections. To ensure that big areas can be processed using a single machine learning model without the need for data reprocessing, data analysis software has to enforce a unique timeline for all tiles.

The differences between between ARD image collections and data cubes proper have led some experts to develop tools that reprocess collections, making them regular in space and in time, and account for missing or noisy values. For example, the Brazil Data Cube provides organised collections (FERREIRA et al., 2020a). The *R gdalcubes* package supports generating consistent data cubes (APPEL; PEBESMA, 2019). When designing *sits* , the authors decided that the software would support both kinds of imagery: regular data cubes and irregular image collections. The design decisions for *sits* will be further explored in the next section.

Figure 2.2 - Sentinel-2 image colour composites for MGRS tile 20LKP in different dates.



2.3 Software design and analysis methods

2.3.1 User requirements

The target audience for *sits* is the new generation of specialists who understand the principles of remote sensing and can write scripts in *R* . To allow experts to use the full extent of available satellite imagery, *sits* adopts a *time-first, space-later* approach. Satellite image image time series are used as inputs to machine learning classifiers; the results are then post-processed using spatial smoothing. Since machine learning methods need accurate training data, *sits* includes methods for quality assessment of training samples. The software also provides tools for model validation and accuracy measurement. The package thus comprises a production environment for big EO data analysis.

Further requirements come from the authors' affiliation with Brazil's National Institute for Space Research (INPE). The institute provides the official Brazilian estimates of deforestation and land use change in the environmental-sensitive Amazonia and Cerrado biomes. Given the emissions and biodiversity impacts of land use change in Amazonia and Cerrado, INPE has been providing estimates of deforestation since 1998. Since 2007, INPE also produces daily alerts of forest cuts (SHIMABUKURO et al., 2012). Comprehensive assessments have shown the quality of INPE's work (PARENTE et al., 2021). Since INPE experts aim to use *sits* to generate monitoring products (FERREIRA et al., 2020a), the package has been designed to meet the performance needs of operational activities.

These requirements led to the following design goals:

a) Encapsulate the land classification workflow in a concise API.

b) Provide access to data cubes and image collections available in cloud services.

c) Develop methods for quality control of training data sets.

d) Offer a single interface to different machine learning and deep learning algorithms.

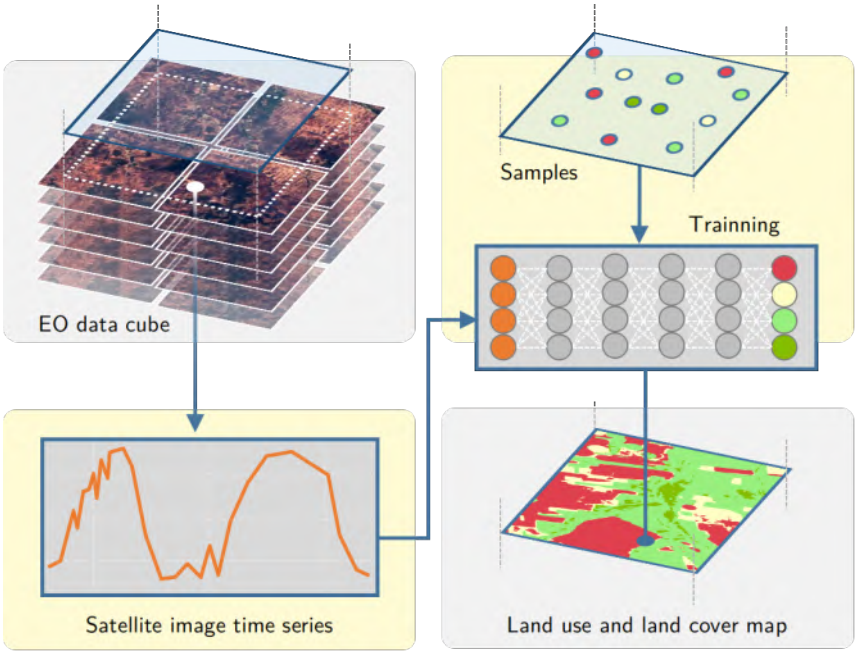
e) Support efficient processing of large areas, with internal support for parallel processing.

f) Include innovative methods for spatial post-processing.

2.3.2 Workflow and API

The design of the *sits* API considers the typical workflow for land classification using satellite image time series (see Figure 2.3). Users define a data cube by selecting a subset of an ARD image collection. They obtain the training data from a set of points in the data cube whose labels are known. After performing quality control on the training samples, users build a machine learning model and use it to classify the entire data cube. The results go through a spatial smoothing phase that removes outliers. Thus, *sits* supports the entire cycle of land use and land cover classification.

Figure 2.3 - Using time series for land classification. EO data cube catalogues are queried (upper-left) and the images data retrieved to form satellite image time series (bottom-left); labelled samples referring to geographic locations and time intervals are used to recover time series that train machine learning models (upper-right); these models are used to classify all time series of EO data cube (bottom-right).



When designing the *sits* API, we tried to capture the essential properties of good software. As stated by Bloch (BLOCH, 2006), good APIs *"should be easy to use and hard to misuse, and should be self-documenting"*. Bloch (BLOCH, 2006) also recommends: *"good programs are modular, and inter-modular boundaries define APIs"*. Following this advice, in *sits* each function carries out one task of the land classification workflow. For example, instead of having separate functions for working

with machine learning models, there is one function for model training. The `sits_train()` function encapsulates all of the differences between different methods, ranging from random forests to convolutional neural networks. All functions have convenient default parameters. Thus, novice users can achieve good results, while more experienced ones are able to fine-tune their models to get further improvements.

The *sits* API captures the main steps of the workflow. These functions are: (a) `sits_cube()` creates a cube; (b) `sits_get_data()` extracts training data from the cube; (c) `sits_train()` trains a machine learning model; (d) `sits_classify()` classifies the cube and produces a probability cube; (e) `sits_smooth()` performs spatial smoothing using the probabilities; (f) `sits_label_classification()` produces the final labelled image. Since these functions encapsulate the core of the package, scripts in *sits* are concise and easy to reuse and reproduce.

2.3.3 Handling data cubes

The *sits* package works with ARD image collections available in different cloud services such as AWS, Microsoft, and Digital Earth Africa. It accepts ARD image collections as input and has user-transparent internal functions that enforce the properties of data cubes (Definitions [1-3]). Currently, *sits* supports data cubes available in the following cloud services: (a) Sentinel-2/2A level 2A images in AWS and on Microsoft's Planetary Computer; (b) collections of Sentinel, Landsat, and CBERS images in the Brazil Data Cube (BDC); (c) collections available in Digital Earth Africa; (d) data cubes produced by the *gdalcubes* package (APPEL; PEBESMA, 2019); (f) local files.

The big EO data sets available in cloud computing services are constantly being updated. For this reason, *sits* uses the STAC (SpatioTemporal Asset Catalogue) protocol. STAC is a specification of geospatial information adopted by providers of big image collections (HANSON, 2019). Using STAC brings important benefits, as *sits* does not need to build indexes of data available in repositories. Relying on STAC web services allows *sits* to access up-to-date information.

Using *sits* , the user defines a data cube by selecting an ARD image collection and determining a space-time extent. Listing 2.1 shows the definition of a data cube using AWS Sentinel-2/2A images. The user selects the "Sentinel-2 Level 2" collection in the AWS cloud service. The data cube's geographical area is defined by the tile

"20LKP" and the temporal extent by a start and end date. Access to other cloud services works in similar ways. Data cubes in *sits* contain only metadata; access to data is done on as-need basis.

Listing 2.1 - Defining a data cube in sits

```
s2_cube <- sits_cube(
    source = "AWS",
    name = "T20LKP_2018_2019",
    collection = "sentinel-s2-l2a",
    tiles = c("20LKP","20LLP"),
    start_date = "2018-07-18",
    end_date = "2018-07-23",
    s2_resolution = 20
)
```

2.3.4 Handling time series

Following the approach taken by the *sf* *R* package for handling geospatial vector objects (PEBESMA, 2018), *sits* stores time series in an object-relational table. As shown in Figure 2.4, a *sits* time series table contains data and metadata. The first six columns contain the metadata: spatial and temporal information, the label assigned to the sample, and the data cube from where the data has been extracted. The spatial location is given in longitude and latitude coordinates for the WGS84 ellipsoid. For example, the first sample at location (−55.2, −10.8) has been labelled "Pasture", being valid during the interval (2013-09-14, 2014-08-29). The `time series` column contains the time series data for each spatiotemporal location.

Figure 2.4 - Data structure for time series.

```
#> # A tibble: 3 x 7
#>   longitude latitude start_date end_date   label   cube    time_series
#>       <dbl>    <dbl> <date>     <date>     <chr>   <chr>   <list>
#> 1     -55.2   -10.8  2013-09-14 2014-08-29 Pasture MOD13Q1 <tibble [23 × 5]>
#> 2     -57.8    -9.76 2006-09-14 2007-08-29 Pasture MOD13Q1 <tibble [23 × 5]>
#> 3     -51.9   -13.4  2014-09-14 2015-08-29 Pasture MOD13Q1 <tibble [23 × 5]>
```

Time series tables store training data used for land use and land cover classification. They are built in two steps. First, experts provide samples with valid locations, labels, and dates, based on field observations or by interpreting high-resolution images. Such

data can be provided as comma-separated text files or as *shapefiles*. Then, *sits* uses the expert data to retrieve the values of time series for each location from the data cube, as illustrated in Listing 2.2.

Listing 2.2 - Extracting time series from a a data cube.

```
# text file containing sample information
csv_file <- "/home/user/samples.csv"
# obtain time series
samples <- sits_get_data(
    cube = s2_cube,
    file = csv_file
)
```

2.3.5 Sample quality control

Experience with machine learning methods shows that the limiting factor in obtaining good results is the number and quality of training samples. Large and accurate data sets are better, no matter the algorithm used (MAXWELL et al., 2018; NOI; KAPPAS, 2018), while noisy and imperfect samples have a negative effect on classification performance (FRENAY; VERLEYSEN, 2014). Software that uses machine learning for satellite image analysis needs good methods for sample quality control.

The *sits* package provides an innovative sample quality control technique based on self-organising maps (SOM) (SANTOS et al., 2019; SANTOS et al., 2021). SOM is a dimensionality reduction technique. High-dimensional data is mapped into two dimensions, keeping the topological relations between similar patterns (KOHONEN, 1990). The input data for quality assessment is a set of training samples, obtained as described in the "Handling Time Series" subsection above. When projecting a high-dimensional data set of training samples into a 2D self-organising map, the units of the map (called “neurons”) compete for each sample. It is expected that good quality samples of each label should be close together in the resulting map. The neighbours of each neuron of a SOM map provide information on intraclass and interclass variability.

The function `som_map()` creates a SOM to assess the quality of the samples. Each sample is assigned to a neuron based on similarity. After the samples are mapped to neurons, each neuron will be associated with a discrete probability distribution. Usually, homogeneous neurons (those with a single label) contain good quality samples. Heterogeneous neurons (those with two or more labels with significant probability) are likely to contain noisy samples. The `som_map()` function provides

quality information for every sample. It also generates a 2D map that is useful to visualise class noise, since neurons associated to the same class are expected to form a cluster in the SOM map.

Figure 2.5 - SOM map for Cerrado training samples.

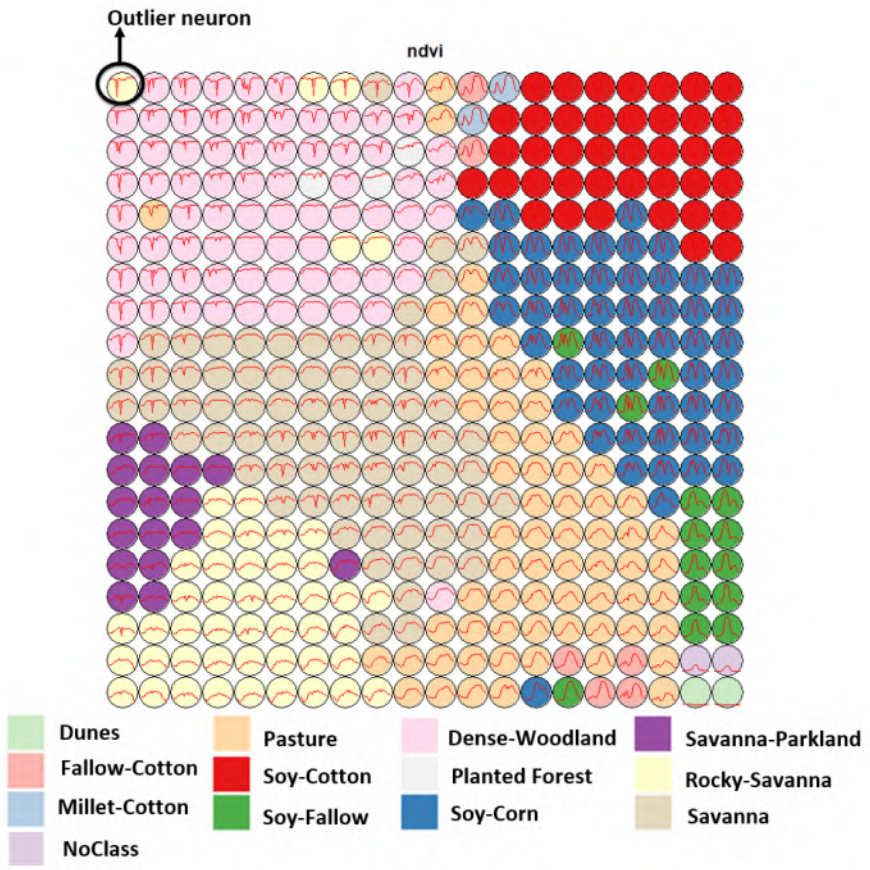


Figure 2.5 shows a SOM map for a set of training samples in the Brazilian Cerrado, obtained from the MODIS MOD13Q1 product. This set ranges from 2000 to 2017 and includes 50,160 land use and land cover samples divided into 12 labels: Dunes, Fallow-Cotton, Millet-Cotton, Soy-Corn, Soy-Cotton, Soy-Fallow, Pasture, Rocky Savanna, Savanna, Dense Woodland, Savanna Parkland, and Planted Forest. Visual inspection shows several outlier neurons located far from their label cluster. For example, while the neurons associated to the "Pasture" label form a cluster, some of those linked to the "Rocky Savanna" label are mixed among those labelled "Dense Woodland", an unexpected situation. The quantitative evaluation confirms this intuitive insight. As shown by Santos et al. (SANTOS et al., 2021), removing these and other outliers improves classification results.

2.3.6 Training machine learning models

After selecting good quality samples, the next step is to train a machine learning model. The package provides support for the classification of time series, preserving the full temporal resolution of the input data. It supports two kinds of machine learning methods. The first group of methods does not explicitly consider spatial or temporal dimensions; these models treat time series as a vector in a high-dimensional feature space. From this class of models, *sits* includes random forests (BELGIU; DRAGUT, 2016), support vector machines (MOUNTRAKIS et al., 2011), extreme gradient boosting (CHEN; GUESTRIN, 2016), and multi-layer perceptrons (PARENTE et al., 2019). The authors have used these methods with success for classifying large areas (PICOLI et al., 2018; SIMOES et al., 2020; PICOLI et al., 2020; FERREIRA et al., 2020a). Our results show that, given good quality samples, *sits* can achieve high classification accuracy using feature space machine learning models.

The second group of models comprises deep learning methods designed to work with image time series. Temporal relations between observed values in a time series are taken into account. Time series classification models for satellite data include 1D convolution neural networks (1D-CNN) (PELLETIER et al., 2019; FAWAZ et al., 2020), recurrent neural networks (RNN) (RUSSWURM; KORNER, 2018), and attention-based deep learning (GARNOT; LANDRIEU, 2020; RUSSWURM et al., 2020). The *sits* package supports a set of 1D-CNN algorithms: TempCNN (PELLETIER et al., 2019), ResNet (FAWAZ et al., 2019), and InceptionTime (FAWAZ et al., 2020). Models based on 1D-CNN treat each band of an image time separately. The order of the samples in the time series is relevant for the classifier. Each layer of the network applies a convolution filter to the output of the previous layer. This cascade of convolutions captures time series features in different time scales (PELLETIER et al., 2019). In the Results section, we show the use of a TempCNN model to classify the Cerrado biome in Brazil for the year 2018.

2.3.7 Data cube classification

The *sits* package runs in any computing environment that supports *R* . When working with big EO data, the target environment for *sits* is a virtual machine located close to the data repository. To achieve efficiency *sits* implements its own parallel processing. Users are not burdened with the need to learn how to do multiprocessing and thus their learning curve is shortened.

Memory management in *R* is a hard challenge. Some advanced machine learning and deep learning methods require dedicated environments outside *R* . For example, the *keras* package in R that supports deep learning relies on *Python* code that calls the *TensorFlow* library written in *C++*. All of these dependencies cause *R* not to have a predictable memory allocation behaviour when doing parallel processing. Given this situation, we developed a customised parallel processing implementation for *sits* to work well with big EO data.

After many tests with different *R* packages that provide support for parallel processing, we found out that no current *R* package meets our needs. The authors implemented a new fault tolerant multi-tasking procedure for big EO data classification. Image classification in *sits* is done by a cluster of independent workers linked to one or more virtual machines. To avoid communication overhead, all large payloads are read and stored independently; direct interaction between the main process and the workers is kept at a minimum. The customised approach, depicted in Figure 2.6, takes the steps presented below.

a) Based on size of the cube, the number of cores, and the available memory, divide the cube into chunks.

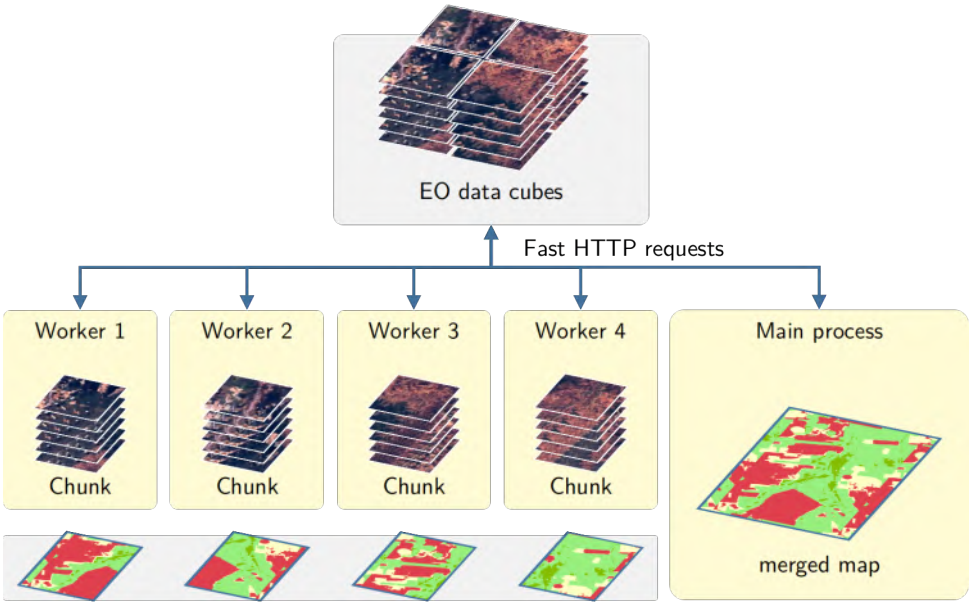
b) Assign chunks to the worker cores. Each core processes a block and produces an output image which is a subset of the result.

c) After all the subimages are produced, join them to obtain the result.

d) If a worker fails to process a block, provide failure recovery and ensure the worker completes the job.

This approach has many advantages. It works in any virtual machine that supports *R* and has no dependencies on proprietary software. Processing is done in a concurrent and independent way, with no communication between workers. Failure of one worker does not cause failure of the big data processing. The software is prepared to resume classification processing from the last processed chunk, preventing against failures such as memory exhaustion, power supply interruption, or network breakdown. From an end-user point of view, all work is done smoothly and transparently.

Figure 2.6 - Parallel processing in *sits* . Main process start workers' process (small yellow areas) that request independently chunks of data from EO data cube through partial data HTTP requests (arrows); each process has a copy of the machine learning model and classifies its chunk (bottom); the final map is obtained by main process by merging classified chunks (right yellow area).



2.3.8 Post-processing

When working with big EO data sets, there is a considerable degree of data variability in each class. As a result, some pixels will be misclassified. These errors are more likely to occur in transition areas between classes or when dealing with mixed pixels. To offset these problems, *sits* includes a post-processing smoothing method based on Bayesian probability that uses information from a pixel's neighbourhood to reduce uncertainty about its label.

The post-classification smoothing uses the output probabilities of a machine learning algorithm. Generally, we label a pixel p_i as being of class k if the probability of that pixel belonging to to class k is higher than any other probability associated to the pixel. Instead of using these probabilities directly, Bayesian smoothing first performs a mathematical transformation by taking the log of the odds ratio for each pixel:

$$\mathbf{x_i} = \log[p_{i,k}/(1 - p_{i,k})] \tag{2.1}$$

To allow mathematical tractability, we assume that $\mathbf{x_i}$ follows a multivariate normal distribution $\mathcal{N}_k(\boldsymbol{\theta}_i, \boldsymbol{\Sigma}_i)$ where k is the number of classes. This distribution has an unknown mean $\boldsymbol{\theta}_i$ and an estimated *a priori* covariance matrix $\boldsymbol{\Sigma}_i$ that controls the level of smoothness to be applied. The covariance matrix represents our prior belief in the class variability and possible confusion between classes.

The local uncertainty is modelled by a multivariate normal distribution $\mathcal{N}_k(\mathbf{m}_i, \mathbf{S}_i)$ where k is the number of classes. The distribution has mean $\mathbf{m}_i$ and covariance matrix $\mathbf{S}_i$. Our strategy to reduce local uncertainty is to estimate these parameters from the neighbourhood of pixel p_i. Taking $\mathbf{X_i}$ as the set of all $\mathbf{x_i}$ vectors in a neighbourhood, we compute $\mathbf{m_i} = \mathrm{E}[\mathbf{X_i}]$ and $\mathbf{S}_i = \mathrm{cov}[\mathbf{X_i}, \mathbf{X_i}]$. The point estimator $\hat{\boldsymbol{\theta}}_i$ for each pixel p_i that minimises the quadratic loss functions can be obtained by applying Bayes' rule. The posterior estimator for the pixel's probabilities can be expressed as

$$\hat{\boldsymbol{\theta}}_{\boldsymbol{i}} = \mathrm{E}[\boldsymbol{\theta}_i|\mathbf{x_i}] = \boldsymbol{\Sigma}_i \left(\boldsymbol{\Sigma}_i + \mathbf{S_i}\right)^{-1} \mathbf{m_i} + \mathbf{S_i} \left(\boldsymbol{\Sigma}_i + \mathbf{S_i}\right)^{-1} \mathbf{x_i}. \tag{2.2}$$

This estimator is computed for each pixel, producing a smoothed map. It is a weighted combination of $\mathbf{x}_i$ and the neighbourhood mean $\mathbf{m}_i$, where the weights are determined by the covariance matrices $\boldsymbol{\Sigma}_i$ and $\mathbf{S_i}$. The component $\left(\boldsymbol{\Sigma}_i + \mathbf{S_i}\right)^{-1}$ plays a normalisation role. Given that the smoothing factor $\boldsymbol{\Sigma}_i$ is provided *a priori* by the user, the estimate depends only on the neighbourhood covariance matrix $\mathbf{S_i}$. When the $\mathbf{X}$ values in a neighbourhood of a pixel are similar, the matrix $\mathbf{S_i}$ increases relative to $\boldsymbol{\Sigma}_i$. In this case, we will have more confidence in the original pixel value and less confidence in the neighbourhood mean $\mathbf{m}$. Likewise, when the $\mathbf{X}$ values in a neighbourhood of a pixel are diverse, the values of the correlation matrix will be low. Thus, the weight expressed by $\mathbf{S_i}$ will decrease relative to $\boldsymbol{\Sigma}_i$. We will have less confidence to the original pixel value $\mathbf{x_i}$ and more confidence in the local mean $\mathbf{m_i}$. The smoothing procedure is thus most relevant in situations where the original classification odds ratio is low, indicating a low level of separability between classes. In these cases, the updated values of the classes will be influenced by the local class variance. The resulting smoothed map will thus consider the influence of the neighbours only when the confidence in the most likely label for a pixel is low. The Bayesian smoothing is a suitable procedure for post-processing probability maps resulting from machine learning methods (SIMOES et al., 2020).

2.3.9 Validation and accuracy assessment

The *sits* package offers support for cross-validation of training models and accuracy assessment of results. Cross-validation estimates the expected prediction error. It uses part of the available samples to fit the classification model, and a different part to test it. The *sits* software does *k-fold* validation. The data is split into k partitions with approximately the same size. The model is tested k times. At each step, *sits* takes one distinct partition for testing and the remaining $k - 1$ partitions for training the model. The results are averaged to estimate the prediction error. The estimates provided by validation are a "best-case" scenario, since they use only the training samples, which are subject to selection bias. Thus, validation is best used to compare different models for the same training data. Such results must not be used as accuracy measures.

To measure the accuracy of classified images, *sits* provides a function that calculates area-weighted estimates (OLOFSSON et al., 2013; OLOFSSON, 2014). The need for area-weighted estimates arises because land use and land cover classes are not evenly distributed in space. In some applications (e.g., deforestation) where the interest lies in assessing how much has changed, the area mapped as deforested is likely to be a small fraction of the total area. If users disregard the relative importance of small areas where change is taking place, the overall accuracy estimate will be inflated and unrealistic. For this reason, the `sits_accuracy_area()` function adjusts the mapped areas to eliminate bias resulting from classification error. This function provides error-adjusted area estimates with confidence intervals, following the best practices proposed by Olofsson et al. (OLOFSSON et al., 2013; OLOFSSON, 2014).

2.3.10 Extensibility

Since one of design aims of *sits* is to keep a simple application programming interface, it uses the *R* S3 object model, which is easily extensible. The designers gave particular attention to the support required for machine learning researchers to include new models in *sits* . For machine learning models, *sits* uses two *R* constructs. In *R* , classifiers should provide a `predict` function, which carries out the actual assignment of input to class probabilities. *R* also provides support for closures, which are functions written by functions (WICKHAM, 2019a). Using closures is particularly useful for dealing with machine learning functions that have completely different internal implementations. The `sits_train()` function in *sits* is a closure that encapsulates the details of how the classifier works. The closure returns a function to classifies time series and data cubes using the overloaded *R* `predict` function. Therefore, training

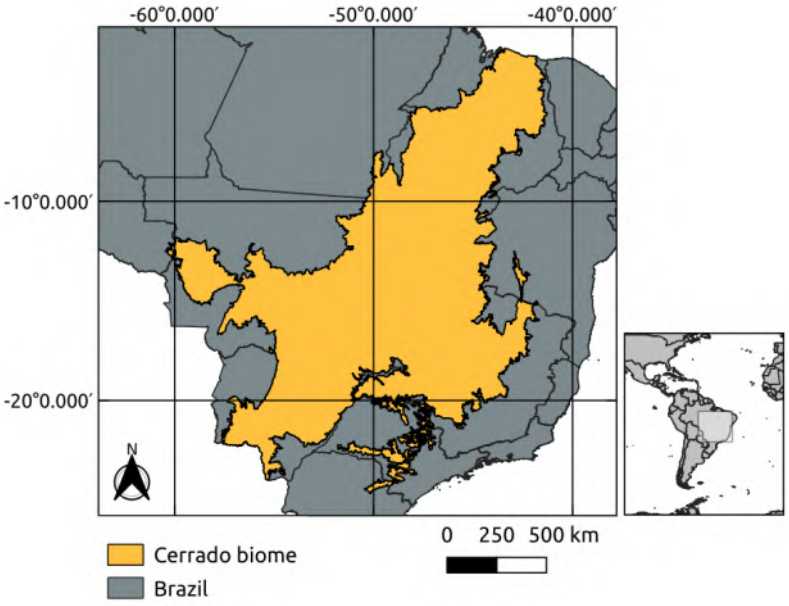
and classification in *sits* are independent and extensibility is easy. Users can provide new models without any need for changing the other components of the package. We expect that both the authors and other contributors to the package will include further advanced deep learning models tailored for image time series.

2.4 Results

In this section, we present an application of *sits* for generation of land cover products. The case study takes a one-year classification of the Cerrado biome in Brazil using Landsat-8 images. Cerrado is the second largest biome in Brazil with 200 million ha (Figure 2.7). It has great biodiversity value due to a large variety of endemic species. Its natural areas are being converted to agriculture at a fast pace, as it is one of the world's fast moving agricultural frontiers (WALTER, 2006).

The Brazilian Cerrado is comprised of a wide diversity of natural vegetation, ranging from grasslands to woodlands. There are three major types of Cerrado physiognomy: "Open Cerrado", typically composed of grasses and small shrubs with a sporadic presence of small tree vegetation; "Cerrado" *Sensu Stricto*, a typical savanna formation, characterized by the presence of low, irregularly branched, thin-trunked trees; and finally, the "Cerradão", a dry forest of medium-sized trees (up to 12-meters), in whose understory it is possible to find grasses and small shrubs (GOODLAND, 1971; DEL-CLARO; TOREZAN-SILINGARDI, 2019).

Figure 2.7 - Cerrado biome extension. Its area covers about 200 million ha.



2.4.1 Input data

The Brazilian Cerrado is covered by 51 Landsat-8 tiles available in the Brazil Data Cube (BDC) (FERREIRA et al., 2020b). Each Landsat tile in the BDC covers an area of 336×220 km, with $11,204 \times 7,324$ pixels in Albers equal area projection. The analysis used a one-year classification period from September 2017 to August 2018, following the agricultural calendar. The temporal interval is 16-days, thus resulting in 24 images per year. We use seven spectral bands plus two vegetation indexes (NDVI and EVI) and the cloud cover information. The total data size for processing is about 8 TB.

2.4.2 Training samples

We did a systematic sampling by defining a grid of points at every 5-kilometers throughout the Cerrado biome, with a total of 85,026 points. The training data labels were extracted using existing land cover products: the pastureland map of 2018 from Pastagem.org (PARENTE et al., 2019), MapBiomas collection 5 for year 2018 (SOUZA et al., 2020), and Brazil's National Mapping Agency IBGE maps for 2016-2018 (INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA - IBGE, 2020).

We retrieved the labels for these $\sim$85K points from each of the maps. For this purpose, we used the land use and land cover trajectory extraction service (WLTS) of the Brazil Data Cube (FERREIRA et al., 2020a). We accepted as valid only points where the labels were the same in the three products. As a result, we got 48,850 points from which we extracted the time series from the Landsat 8 data cube. The distribution of the samples for each class is the following: "Annual Crop" (6,887), "Cerradao" (4,211), "Cerrado" (16,251), "Natural Non Vegetated" (38), "Open Cerrado" (5,658), "Pasture" (12,894), "Perennial Crop" (68), "Silviculture" (805), "Sugarcane" (1775), and "Water" (263).

2.4.3 Training and classification

Our maps were classified using the TempCNN method (PELLETIER et al., 2019). The model was trained with the $\sim 48K$ training samples described above. All available attributes in the BDC Landsat 8 data cube (2 vegetation indices and 7 spectral bands) were used for training and classification. We used the default configuration of the TempCNN method in *sits* , which has three 1D convolutional layers followed by a multilayer perceptron (PELLETIER et al., 2019). After the classification, we applied Bayesian smoothing to the probability maps and then generated a labelled map,

selecting the most likely class for each pixel. The classification was executed on Ubuntu server, using 24 cores and 128 GB memory. Each Landsat tile was classified in an average of 29 minutes, and the total classification took about 24 hours. Figure 2.8 shows the final map.

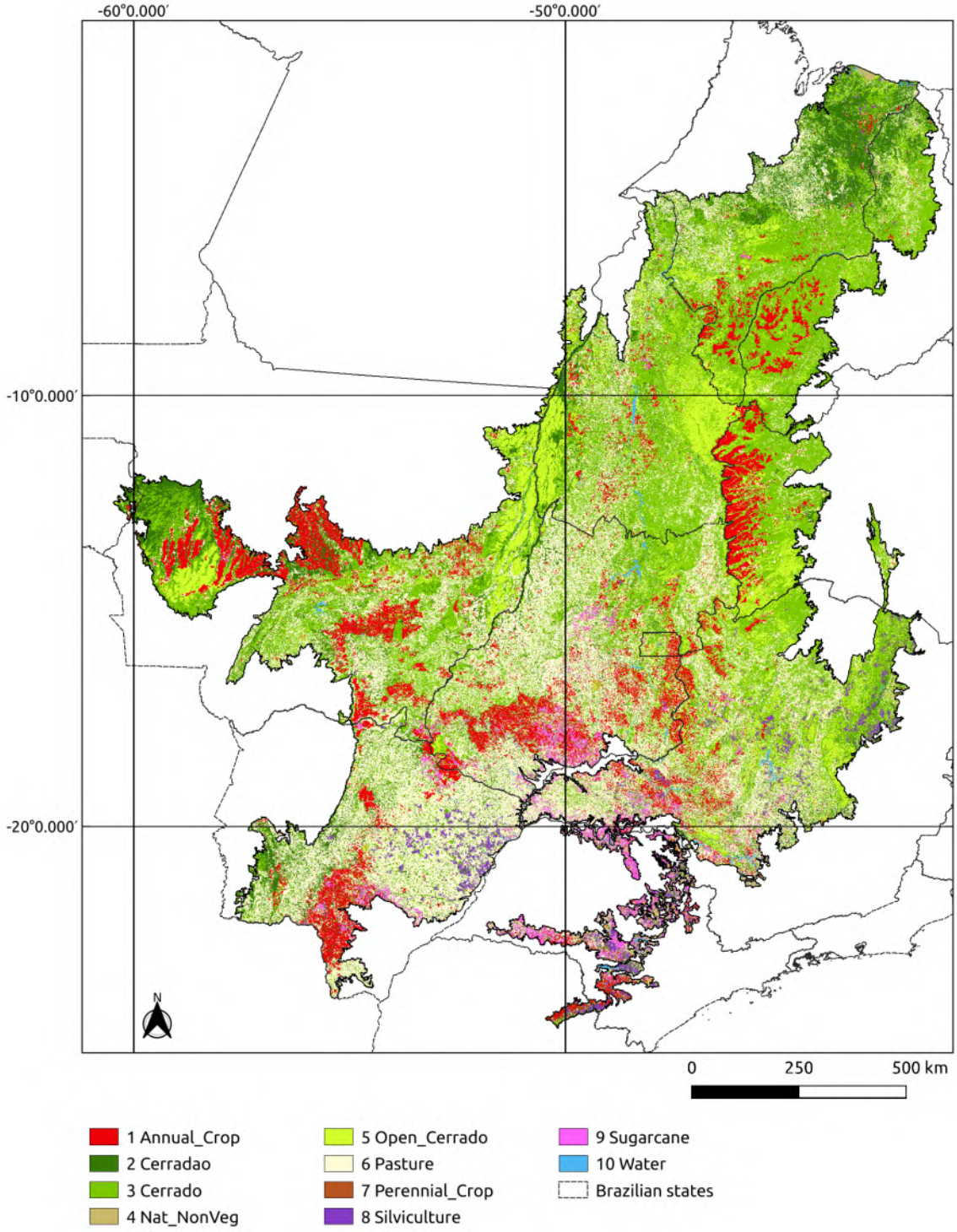
2.4.4 Code

According to the principles of the *sits* API, each function is responsible for one of the workflow tasks. As a consequence, the classification of the whole Cerrado (200 million ha) is achieved by six *R* commands, as shown in Listing 2.3.

Listing 2.3 - R Code for Cerrado LUCC classification.

```
# defines a reference to a data cube stored in the BDC.
cube <- sits_cube(
    source = "BDC",
    name = "cerrado",
    collection = "LC8_30_16D_STK-1",
    roi = "./cerrado.shp", # region of interest is a shapefile
    start_date = "2017-09-01",
    end_date = "2018-08-31"
)
# obtain the time series for the samples
samples <- sits_get_data(
    data = cube,
    file = "./samples.csv" # samples in CSV
)
# train model using TempCNN algorithm
cnn_model <- sits_train(
    data = samples,
    ml_method = sits_TempCNN()
)
# classify data cube
probs <- sits_classify(
    data = cube,
    ml_model = cnn_model
)
# compute Bayesian smoothing
probs_smooth <- sits_smooth(
    cube = probs
)
# generate thematic map
map <- sits_label_classification(
    cube = probs_smooth
)
```

Figure 2.8 - Cerrado land Use and Land Cover map for year 2018. The map was generated using temporal convolutional neural networks algorithm trained with $\sim 48K$ samples extracted from three existing land cover products. A spatial smoothing Bayesian was applied to obtain the final result.



2.4.5 Classification accuracy

An accuracy assessment of the land use and land cover map was conducted by a systematic sampling defined by a grid of 20-kilometers throughout the Cerrado biome, with a total of 5,402 points. These samples are independent of the training set used in the classification. The samples were interpreted by five specialists using high resolution images from the same period of the classification. For the assessment, we merged the labels "Cerradao" "Cerrado" and "Open Cerrado" to one label called "Cerrado". We also did additional sampling to reach a minimal number of samples for the classes "Natural Non Vegetated", "Perennial Crop", and "Water". This resulted in 5,286 evaluation samples thus distributed: "Annual Crop" (553), "Cerradao" (704), "Cerrado" (2451), "Natural Non Vegetated" (44), "Pasture" (1246), "Perennial Crop" (38), "Silviculture" (94), "Sugarcane" (77), and "Water" (79). We used the *sits* implementation of the area-weighted technique (OLOFSSON et al., 2013) to provide an unbiased estimator for the overall accuracy and the total area of each land use and land cover class based on the reference samples.

Table 2.1 - Area-weigthed classification accuracy.

Labels	**User's accuracy**	**Producer's accuracy**
Annual Crop	0.88	0.81
Cerrado	0.91	0.89
Natural Non Vegetated	0.95	0.63
Pasture	0.76	0.82
Perennial Crop	0.74	0.51
Silviculture	0.91	0.83
Sugarcane	0.81	0.96
Water	0.97	0.93

Overall Accuracy: 0.86

2.5 Discussion

The good results of mapping the entire Cerrado biome using *sits* show that the software has met its design goals. It was possible to classify a large area of about 200 million ha using an advanced deep learning model on time series with good performance. Running the whole training and classification process requires a script with only 6 *R* commands. No specific knowledge of parallel processing was required. All in all, the concept of having an integrated solution has been demonstrated.

We now discuss some of the thematic results from the mapping of the Cerrado biome. The classes which have the worst performance are "Perennial Crop" and "Natural Non Vegetated" with producer's accuracy of 51% and 63%, respectively. Since these classes are associated with small areas of the Cerrado biome, they had fewer training samples. The authors had access to only 68 samples of the "Perennial Crop" class and 38 samples of the "Natural Non Vegetated" class. To improve the accuracy of these classes, future classifications need to ensure that the number of samples per class is balanced and representative.

Considering the aims and design of *sits* , it is relevant to discuss how its design and implementation choices differ from other software for big EO data analytics, such as Google Earth Engine (GORELICK et al., 2017), Open Data Cube (LEWIS et al., 2017) and textitopenEO (SCHRAMM et al., 2021). In what follows, we compare *sits* to each of these solutions.

Google Earth Engine (GEE) (GORELICK et al., 2017) uses the Google distributed file system (GHEMAWAT et al., 2003) and its map-reduce paradigm (DEAN; GHEMAWAT, 2008). By combining a flexible API with an efficient back-end processing, GEE has become a widely used platform (AMANI et al., 2020). However, GEE is restricted to the Google environment and does not provide direct support for deep learning. By contrast, *sits* aims to support different cloud environments and to allow advances in data analysis by providing a user-extensible interface to include new machine learning algorithms.

The Open Data Cube (ODC) is an important contribution to the EO community and has proven its usefulness in many domains (LEWIS et al., 2017; GIULIANI et al., 2020). It reads subsets of image collections and makes them available to users as a Python *xarray* structure. ODC does not provide an API to work with *xarrays*, relying on the tools available in Python. This choice allows much flexibility at the cost of increasing the learning curve. It also means that temporal continuity is restricted to the *xarray* memory data structure; cases where tiles from an image collection have different timelines are not handled by ODC. The design of *sits* takes a different approach, favouring a simple API with few commands to reduce the learning curve. Processing and handling large image collections in *sits* does not require knowledge of parallel programming tools. Thus, *sits* and ODC have different aims and will appeal to different classes of users.

Designers of the *openEO* API (SCHRAMM et al., 2021) aim to support applications that are both language-independent and server-independent. To achieve their goals, *openEO* designers use microservices based on REST protocols. The main abstraction of *openEO* is a *process*, defined as an operation that performs a specific task. Processes are described in JSON and can be chained in process graphs. The software relies on server-specific implementations that translate an *openEO* process graph into an executable script. Arguably, *openEO* is the most ambitious solution for reproducibility across different EO data cubes. To achieve its goals, *openEO* needs to overcome some challenges. Most data analysis functions are not self-contained. For example, machine learning algorithms depend on libraries such as *TensorFlow* and *Torch*. If these libraries are not available in the target environment, the user-requested process may not be executable. Thus, while the authors expect *openEO* to evolve into a widely-used API, it is not yet feasible to base an user-driven operational software such as *sits* in *openEO*.

Designing software for big Earth observation data analysis requires making compromises between flexibility, interoperability, efficiency, and ease of use. GEE is constrained by the Google environment and excels at certain tasks (e.g., pixel-based processing) while being limited at others such as deep learning. ODC allows users complete flexibility in the Python ecosystem, at the cost of limitations when working with large areas and requiring programming skills. The *openEO* API achieves platform independence but needs additional effort in designing drivers for specific languages and cloud services. While the *sits* API provides a simple and powerful environment for land classification, it has currently no support for other kinds of EO applications. Therefore, each of these solutions has benefits and drawbacks. Potential users need to understand the design choices and constraints to decide which software best meets their needs.

2.6 Conclusions

The development of analytical software for big EO data faces several challenges. Designers need to balance between conflicting factors. Solutions which are efficient for specific hardware architectures can not be used in other environments. Packages that work on generic hardware and open standards will not have the same performance as dedicated solutions. Software that assumes that its users are computer programmers are flexible, but may be difficult for a wide audience to learn. The challenges lead a

diversity of solutions in academia and industry to work with big Earth observation data. Arguably, it is unlikely that a single approach will emerge as the complete best solution for big EO analytics.

Despite the challenges, there are points of convergence and common ground between most of the solutions for big EO data. The STAC protocol has emerged as a *de facto* standard for describing EO image collections. Users need interoperable and reusable solutions, where the same software can be used in different cloud services with similar results. Experience with existing solutions shows the benefits of simple APIs for the remote sensing community at large. These commonalities should be considered by big EO software designers.

In the design of *sits* , we had to make choices. We took an early option of focusing on time series analysis, based on the hypothesis that time series provide an adequate description of changes in land use and land cover. Instead of relying on time series metrics, we opted to allow machine learning methods to find patterns in multidimensional spaces by providing them all available data. The design of the *sits* data structures and API follows from our choice of doing land classification based on time series analysis.

Plans for evolution of the *sits* package include supporting new deep learning classifiers targeted for satellite image time series analysis. We plan to include manipulation of data cubes, allowing mathematical operations to be performed. Another priority is improving the training phase, using techniques such as active learning and semi-supervised self learning. We also intend to include rule-based post-processing to allow multiyear classification comparison. Given the global applicability of *sits* , we intend to support a user and developer community by providing guidance and documentation.

3 LAND USE AND COVER MAPS FOR MATO GROSSO STATE IN BRAZIL FROM 2001 TO 2017

This Chapter[1] presents a dataset of yearly land cover classification maps for Mato Grosso State, Brazil, from 2001 to 2017. Mato Grosso is one of the world's fast moving agricultural frontiers. To ensure multi-year compatibility, the work uses MODIS analysis-ready products and an innovative method that applies machine learning techniques to classify satellite image time series. The maps provide information about crop and pasture expansion over natural vegetation, as well as spatially explicit estimates of increases in agricultural productivity and trade-offs between crop and pasture expansion. Therefore, the dataset provides new and relevant information to understand the impact of environmental policies on the expansion of tropical agriculture in Brazil. Using such results, researchers can make informed assessments of the interplay between production and protection within Amazon, Cerrado, and Pantanal biomes.

3.1 Introduction

Brazil is one of the top agricultural producers and exporters, being the largest extent of tropical rainforest and home to an estimated 15% to 20% of the world's biodiversity. Such unique position leads to the need for balancing agricultural production and environmental protection (MARTINELLI et al., 2010). Without substantial investments in productivity and strong land policies, the expansion of agricultural production in Brazil can be a significant factor in environmental degradation. It is vital to understand the impact of environmental policies on the expansion of tropical agriculture in Brazil.

In Nationally Determined Contribution (NDC) to the United Nations Framework Convention on Climate Change (UNFCCC) under the 2015 Paris Agreement, Brazil aims for zero illegal deforestation and zero net emissions within the Amazon rainforest by 2030. The forest emission balance will be achieved by restoring and reforesting 12 million hectares. Brazil's NDC also makes a firm commitment to promote low-carbon agriculture and to increase biofuel use for transportation. Overall, achieving the emission reduction goals Brazil set in its NDC will highly depend on how the country meets the targets associated with the land use sector.

[1] A version of this Chapter was published as a data descriptor in the Nature Scientific Data journal, and was co-authored by Michelle C. A. Picoli, Gilberto Camara, Adeline Maciel, Lorena Santos, Pedro R. Andrade, Alber Sánchez, Karine Ferreira, Alexandre Carvalho (SIMOES et al., 2020).

Since the election of current Brazilian president in late 2018, there is a growing tension between the interests of Brazilian agricultural exporters and rural producers mostly linked to extensive cattle ranching. While the export sector supports the country's pledges to the Paris Agreement, most cattle ranchers and smallholders do not want to commit to environmental protection policies (GIBBS et al., 2015). Since the traditional rural sector is one of the primary supporters of current government, there are increasing concerns about whether Brazil will be committed to achieve its NDC. Comparing the environmental impact of different agricultural sectors is therefore important for all those interested in land policies in Brazil.

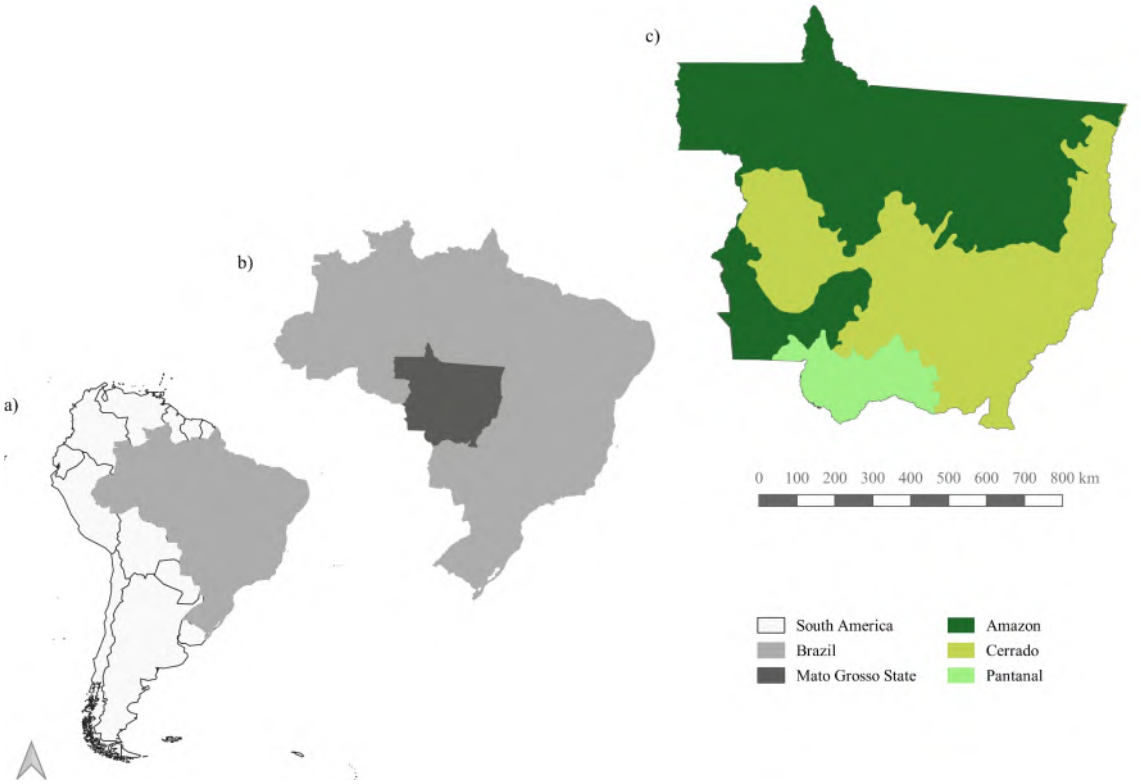
The legal basis for land policies in Brazil is the Forest Code. When created in 1965, it established a proportion of rural properties that must be permanently maintained as forest (legal reserve). It also prohibited clearing vegetation in sensitive areas such as steep slopes and along riverbanks and streams. In 2012, Congress approved a revision of the Forest Code. It stipulates that landowners, in the Legal Amazon, must conserve 80% of their property in forest areas, 35% in cerrado areas, 20% in general fields. To monitor compliance with the new Forest Code, Brazil has been successfully using wall-to-wall satellite-based monitoring (TYUKAVINA et al., 2017). Using satellite observations allows consistent monitoring of land use change, which is necessary for assessing how effective have been the enforcement of public policies as well as the new Forest Code (SOTERRONI et al., 2018).

One particular area of interest for understanding the balance between production and protection in Brazil is the Mato Grosso State, one of the world's most extensive agricultural frontiers (ARVOR et al., 2011; SPERA et al., 2014; KASTENS et al., 2017; PICOLI et al., 2018). Mato Grosso is the third largest state of Brazil, with an area of $90,335,700$ ha. If it was a country, it would be the world's 33^{rd} largest one, being almost as large as Venezuela and Nigeria. Mato Grosso also contains part of three Brazilian biomes: Amazon, Cerrado, and Pantanal (Figure 3.1). From 1988 to 2018, Brazil's National Institute for Space Research (INPE) estimates that 14.5 million ha of natural forests in the Amazonia biome in Mato Grosso have been clear-cut. INPE also estimates that, from 2001 to 2018 in Mato Grosso, more 4.5 million ha of natural cerrado vegetation have been removed.

Based on the above motivation, this Chapter describes a data set of yearly land use and land cover maps for Mato Grosso from 2001 to 2017. These maps are temporally consistent and provide information on deforestation and changes in natural vegetation, crop and pasture expansion, a well as productivity increase. To ensure multi-year

compatibility, the work uses the Moderate Resolution Imaging Spectroradiometer (MODIS) sensor analysis-ready products and an innovative method that applies machine learning techniques to classify satellite image time series. Using the results, it is possible to make informed assessments of the interplay between production and protection in the Amazon, Cerrado, and Pantanal biomes.

Figure 3.1 - Mato Grosso study area: a) location of Mato Grosso relative to South America continent; b) location of Mato Grosso State relative to Brazil; c) Mato Grosso State biomes.



3.2 Methods

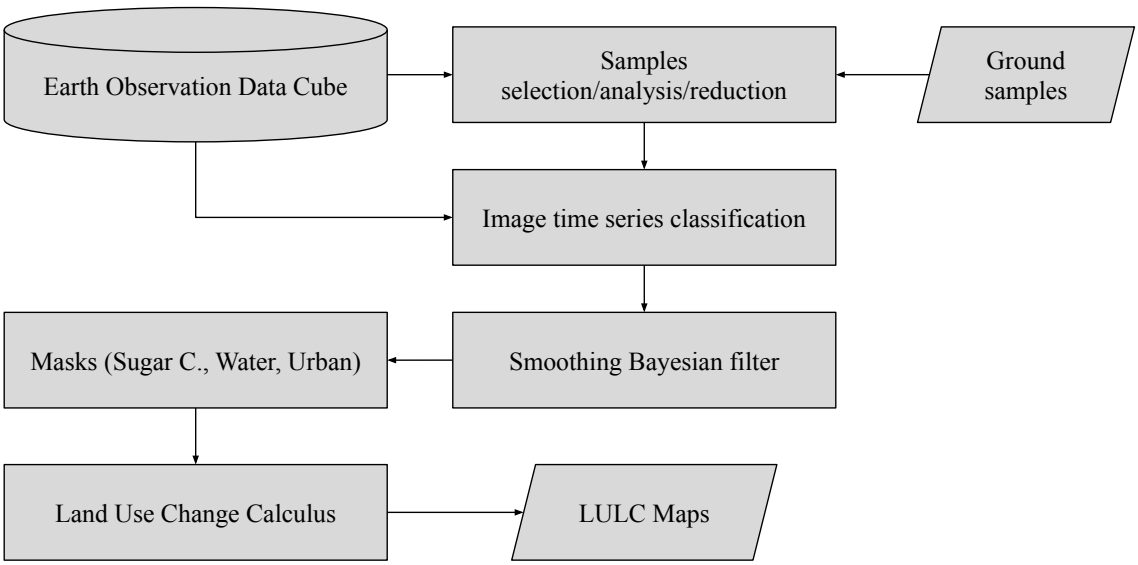
In this section, we detail our approach to generate land use and cover maps of Mato Grosso State. The main steps are depicted in Figure 3.2.

3.2.1 Input data

We based our work on Earth observation data cube, using data stored in a cloud service (Amazon Web Services), over which we ran the classification. The input data is a set of MOD13Q1 collection 6 images, provided by NASA/LPDAAC from

2000-09-01 to 2017-08-31, covering Mato Grosso State. The MOD13Q1 images are available every 16 days at a 250-meter spatial resolution in the sinusoidal projection (DIDAN, 2015). For our analysis, we used normalised difference vegetation index (NDVI), enhanced vegetation index (EVI), near-infrared (NIR), and mid-infrared (MIR) attributes.

Figure 3.2 - Diagram depicting our methods.



The samples data set has 2,115 samples containing longitude, latitude, start date, end date, and label. We defined nine land use and cover classes: (1) "Forest", (2) "Cerrado", (3) "Pasture", (4) "Soy-Fallow" (single crop), (5) "Fallow-Cotton" (single crop), (6) "Soy-Cotton" (double crop), (7) "Soy-Corn" (double crop), (8) "Soy-Millet" (double crop), and (9) "Soy-Sunflower" (double crop). Our samples range from 2000 to 2015 and all samples are available at PANGAEA repository (CAMARA et al., 2019). The crop and pasture ground data was collected through field observations and farmer interviews provided by (KASTENS et al., 2017; SANCHES et al., 2018). Samples for "Cerrado" and "Forest" were provided through fieldwork and high-resolution images. Ground samples for "Soybean-Fallow" were provided through fieldwork, based on previous work of (ARVOR et al., 2011). The classes are shown in Table 3.1.

We retrieved time series data of the 2,115 samples from the Web Time Series Service (WTSS) (VINHAS et al., 2016), an R package available on CRAN (https://CRAN.R-project.org/package=wtss)[2]. Each sample corresponds to one year of observations that comprises 23 values of MOD13Q1 per band.

Table 3.1 - Samples used for training the classification model.

Class label	Count	Frequency
Cerrado	379	20.0%
Fallow-Cotton	29	1.5%
Forest	131	6.9%
Pasture	344	18.2%
Soy-Corn	364	19.2%
Soy-Cotton	352	18.6%
Soy-Fallow	87	4.6%
Soy-Millet	180	9.5%
Soy-Sunflower	26	1.4%

The temporal patterns of the ground samples (Table 3.1), using NDVI, EVI, NIR, and MIR bands, can be seen in Figure 3.3, which uses a generalised additive model to estimate the joint distribution of the samples data set for each class (MAUS et al., 2016).

3.2.2 Pre-processing for sample quality control

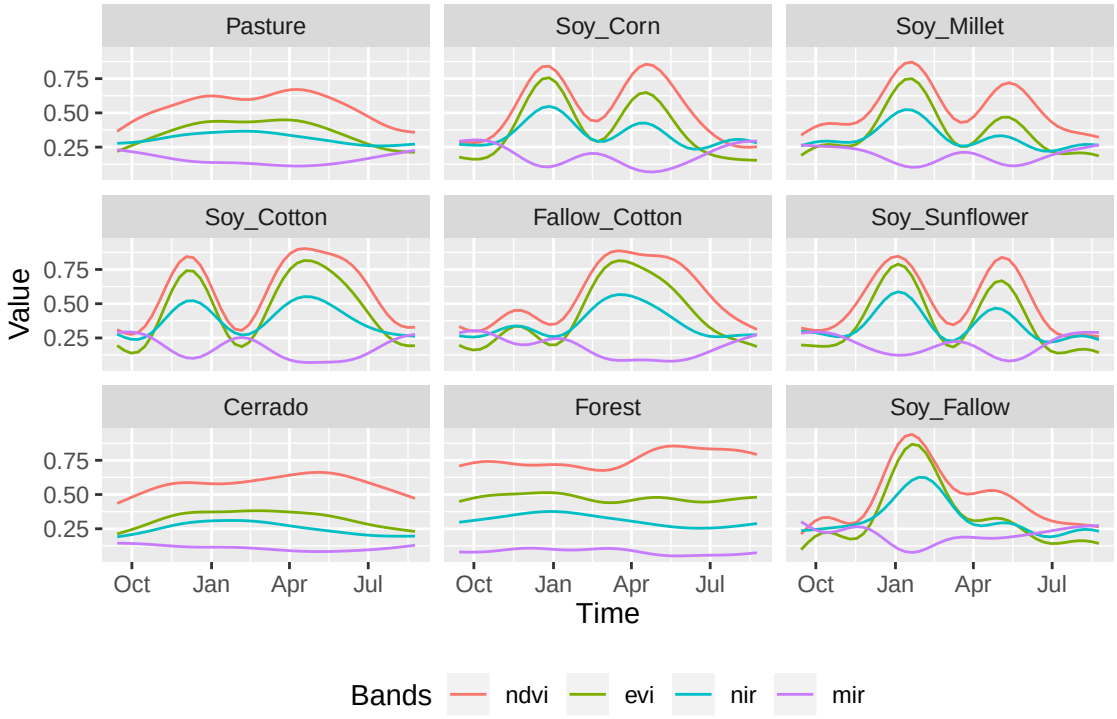
One of the key challenges when using samples to train machine learning classification models is assessing their quality. Noisy and imperfect training samples can have a negative effect on classification performance (FRENAY; VERLEYSEN, 2014). Therefore, it is useful to apply pre-processing methods to improve the quality of the samples and to remove those that might have been wrongly labelled or that have low discriminatory power. In this work, we applied a clustering method based on self-organizing maps (SOM) neural network to test sample quality.

Self-organizing maps is a dimensionality reduction technique (KOHONEN, 1990). High-dimensional data is mapped into two dimensions, keeping the topological relations between data patterns. This allows users to visualise and assess the structure of the input data set. For quality control of the training data, we used a SOM clustering

[2]Author note: by the time of this thesis edition, the WTSS package is not anymore listed in The comprehensive R archive network (CRAN)

method, whose input is a time series samples data set. The output layer comprises a 2D grid of neurons, each associated to a weight vector of the same dimension as the input space.

Figure 3.3 - Temporal patterns of NDVI, EVI, NIR, and MIR bands for the land use and cover classes. The patterns are obtained using a generalised additive model on time series of samples data set.



Source: Picoli et al. (2018)

In the SOM algorithm, the 2D grid of neurons is initialised randomly. Then, for each time series sample, the algorithm finds the neuron with the smallest distance to the sample, based on its weight vector. After the match, the neuron's weight vector and those of its neighbours are then updated. After all training samples are associated with neurons, each neuron is labelled using a majority vote, taking the most frequent class from the samples associated with it. In this way, SOM splits the output space into regions. It is expected that good quality samples of each class would be close together in the resulting map.

To increase the reliability of quality control procedures, SOM was executed several times. Using this iterative procedure, we computed the probability of each sample belonging to the resulting clusters. From these probabilities, we analysed the separability of samples with similar phenological patterns and decided which samples were to be discarded. This allows using SOM to detect and remove outliers. Table 3.2 illustrates how this method works. It contains three samples of "Pasture", identified from 1 to 3. For sample 1, its original and clustered label matched 100% of the time. For sample 2, only 52% of the original and cluster labels matched. Finally, sample 3 matched only 5% of the original label. We define good samples as those that match at least 80% of the time. Thus, in this example, samples 2 and 3 were removed from the data set.

Table 3.2 - Reliability of three pasture samples, identified as 1, 2, and 3.

Identifier	Original class	Cluster label	Frequency
1	Pasture	Pasture	100%
2	Pasture	Pasture	52%
		Cerrado	41%
		Forest	2%
		Soy-Corn	2%
		Soy-Cotton	2%
		Fallow-Cotton	1%
3	Pasture	Cerrado	94%
		Pasture	5%
		Forest	1%

For Mato Grosso samples, the SOM-based clustering reduced the training data set by 10.5%, from 2,115 to 1,892 entries. This filtered data set was then used to train the classification model.

3.2.3 Image time series classification

To generate our maps, we used support vector machine (SVM) as a classification model. Given a multidimensional data set, SVM finds an optimal separation hyperplane that minimises misclassifications (CORTES; VAPNIK, 1995). For data that is not linearly separable, SVM includes kernel functions that map the original feature space into a higher dimensional space, providing nonlinear boundaries to the original feature space. SVM is one of the most widely used algorithms in machine learning applications and has been widely applied to classify remote sensing data (MOUNTRAKIS et al., 2011).

In a recent review of machine learning methods to classify remote sensing data (MAXWELL et al., 2018), the authors note that many factors influence the performance of these classifiers, including the size and quality of the training data set, the dimension of the feature space, and the choice of the parameters. With Mato Grosso data, previous experiments by the authors have shown that the SVM classifier has a better performance than other machine learning methods such as random forest for MOD13Q1 time series data (PICOLI et al., 2018).

For SVM training, we used a 92-dimensional feature space, comprising four time series for each pixel. Each time series contains 23 samples of one of the MOD13Q1 bands NIR, MIR, EVI, and NDVI. The data set of $1,892$ labelled and quality-controlled time series was used to train an SVM model using a radial basis kernel function (RBF), with cost $C = 1$ and $\gamma = 1/92$ (HASTIE et al., 2009). We chose these parameters based on a 5-fold cross-validation test. Parameter C is the cost used as a softened parameter of hyperplane boundaries, while γ is a distance normalisation parameter used in the RBF kernel. We used *sits* *R* package to train SVM and to classify all MOD13Q1 tiles of Mato Grosso stored in an AWS S3 service.

3.2.4 Local smoothing by Bayesian filtering

One of the well-established methods in remote sensing image analysis is to combine pixel-based classification methods with a spatial post-processing method to remove outliers and misclassified pixels. Methods proposed in the literature include modal filters (GHIMIRE et al., 2010) and probabilistic relaxation (GONG; HOWARTH, 1989). Our method uses Bayesian smoothing to reclassify the pixels.

Usually, machine learning methods assign class probabilities to each pixel. Most applications using this approach select the most probable class from the classifier output to be the categorical result for each pixel. The proposed method uses all pixel classes probabilities to compute the resulting confidence. When the magnitude of the discrepancies among the pixel probabilities is high, we have a higher confidence in the classification. Otherwise, when the probabilities have similar magnitudes, we have a low confidence. This is a typical situation in borders and mixed pixels.

To change low confidence pixels, we followed well-established Bayesian smoothing methods (CRESSIE, 1995): borrow strength from the neighbours to reduce the variance of the estimated class for each pixel. The main rationale is to use Bayesian inference considering the mean and variance of the pixel's neighbours to calculate the posterior Bayesian probabilities, then reevaluate the most probable class for the pixel. This

procedure can change the class of pixels with low confidence to the neighbourhood class with a higher confidence. When the local class variance of the neighbours is high, the method gives more weight to the original pixel value. The smoothing algorithm[3] considers a global parameter σ^2 that weighs the smoothness level by increasing the influence of the neighbourhood. When $\sigma^2 = 0$, there is no change in the posterior Bayesian class probabilities. Positive values of σ^2 indicate the influence the neighbours in the posterior Bayesian probabilities.

In our case, after some classification tests, we set $\sigma^2 = 10$ for all classes, which showed the best performance in the technical validation. Additionally, we used a single neighbourhood rule, being all those pixels with Chebyshev distance equal to one, which is the same as to consider a 3×3 window around a pixel.

3.2.5 Post-processing: masks and land use change calculus

Three land cover classes that are not included in the training data set were introduced as masks on all output maps: sugarcane, water, and urban areas. The sugarcane mask from 2003 to 2016 comes from Canasat project (RUDORFF et al., 2010), which maps sugarcane areas in the South-Central region of Brazil using Landsat series images (ADAMI et al., 2012). The water mask comes from (PEKEL et al., 2016), who used three million Landsat series satellite images to quantify changes in global surface water over the past 32 years (1984 to 2015). Finally, the urban area mask was provided by (SPAROVEK et al., 2015).

To prepare the base map (year 2001), we considered using the Deforestation Monitoring Project (PRODES) Amazon (INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS - INPE, 2019b) and PRODES Cerrado (INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS - INPE, 2019a) data sets produced by Brazil's National Institute for Space Research (INPE). We used PRODES datasets to achieve a better consistency in our base map, as our classification method is unaware of preceding land cover trajectories before 2001. These data sets are the official Brazilian statistics on deforestation (SHIMABUKURO et al., 2012). Table 3.3 lists the set of rules applied on the base map using these data sets. We applied each rule comparing corresponding pixels of two

[3]Author note: this description refer to an early version of *sits* Bayesian filter implementation where the inference was taken for each class, independently. In the current version of *sits* this algorithm improved to a vectorial approach of Bayesian inference where all classes probabilities and smoothing parameter are vector and matrices, just like presented in Chapter 2.

maps. We denote M_i, A_i, and C_i as the classes of pixel i of Mato Grosso classification, PRODES Amazon, and PRODES Cerrado maps, respectively. The result of each rule is a new class for M_i, described in the right column.

Table 3.3 - Rules applied on the base map (year 2001).

Condition	Resulting class for M_i
$A_i = \langle Forest \rangle$	$\langle Forest \rangle$
$A_i = \langle Non\text{-}Forest \rangle$ and $M_i = \langle Forest \rangle$	$M_i = \langle Cerrado \rangle$
$A_i = \langle Deforestation \rangle$ and $M_i = \langle Forest \rangle$	$M_i = \langle Sec\text{-}Vegetation \rangle$
$C_i = \langle Non\text{-}Anthropized \rangle$ and $M_i = \langle Forest \rangle$	$M_i = \langle Cerrado \rangle$
$C_i = \langle Anthropized \rangle$ and $M_i = \langle Forest \rangle$	$M_i = \langle Sec\text{-}Vegetation \rangle$

Since our method produces maps independently, the results may have temporal inconsistencies. For example, a natural forest area that was cut in one year may regrow back to forest after being abandoned. In this case, it is useful to distinguish pristine forest from secondary vegetation, which is also classified as a forest in later years. Another example is the case when a pixel classified as "Forest" in one year is classified as "Cerrado" in another year, which represents an impossible transition. To handle these inconsistencies, we used the Land Use Change Calculus (LUC Calculus) (MACIEL et al., 2019) over all produced maps as a set of land use trajectories from 2001 to 2017, using 2001 as a reference date.

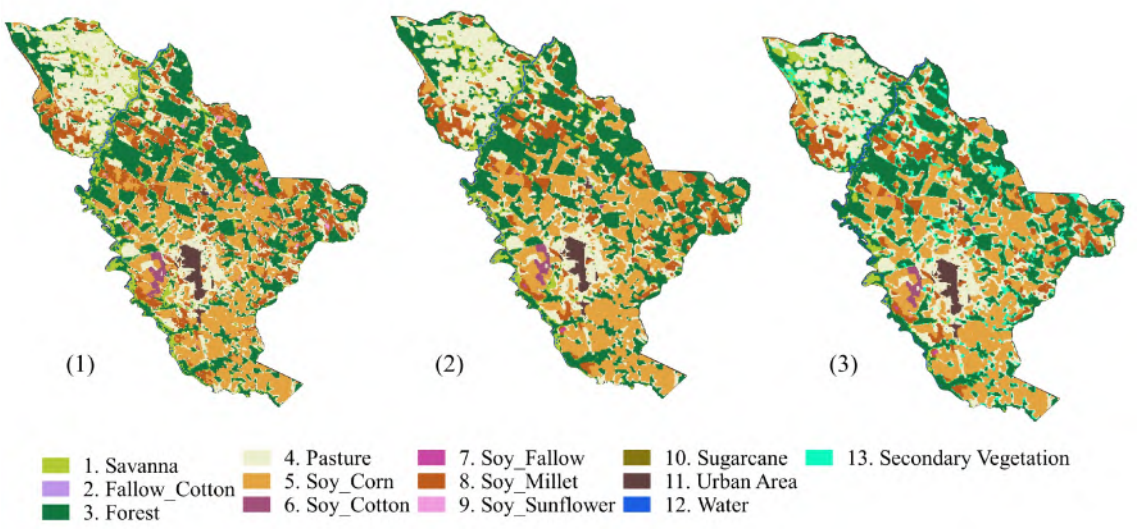
Table 3.4 expresses the set of rules that describe when a particular class will be replaced by another, using LUC Calculus. We considered four classes: "Forest" (F), "Cerrado" (C), "Pasture" (P), and "Soybean" (any class with soybean) (S). The rules were applied sequentially to ensure the temporal consistency among classes over the years. The expressions indicate that trajectories on the left side of '$\rightarrow$' are updated to the classes on the right side. The updated classes are those highlighted with '$*$' symbol. All rules assume 2001 as the base map. Rules nine and ten create a new class for secondary vegetation (SV). This class represents deforested areas that regrew as a secondary forest after being abandoned.

Figure 3.4 displays an example of how post-processing was applied. It shows different results for Sinop municipality in 2016. Figure 3.4-(1) illustrates the original SVM output map, after applying water, sugarcane, and urban masks. Figure 3.4-(2) shows the result after Bayesian smoothing on the original classification. Finally, Figure 3.4-(3) shows the output map after LUC Calculus. This last map is the only one that identifies secondary vegetation areas.

Table 3.4 - Rules applied over all classified years. The classes labels: “Forest” (F), “Cerrado” (C), “Pasture” (P), “Soybean” (S), and “Secondary Vegetation” (SV).

Land-use transition rules		
1. C → F*	⟹	C → C*
2. C → C → P* → C	⟹	C → C → C* → C
3. C → C → S* → C	⟹	C → C → C* → C
4. P → P → C* → C* → P	⟹	P → P → P* → P* → P
5. F → C* → F → F	⟹	F → F* → F → F
6. F → F → C* → F	⟹	F → F → F* → F
7. F → C* → F	⟹	F → F* → F
8. F → C*	⟹	F → F*
9. F → F → P → F*	⟹	F → F → P → SV*
10. P → P → F* → P	⟹	P → P → SV* → P

Figure 3.4 - Detail of SVM classification for Sinop municipality in 2016. (1) Original map with masks; (2) original map with Bayesian smoothing and masks; (3) final map after applying the LUC Calculus and masks.



3.3 Data records

The data set provides annual land use and cover maps from 2001 to 2017 in sinusoidal projection, which is the same cartographical projection used by the input MODIS images. The archive available at PANGAEA (CAMARA et al., 2019) contains the classified maps in compressed TIFF format (one per year) at MODIS resolution, as well as a file with the training data set (1,892 ground samples) in CSV format and a style file for displaying the data in QGIS.

3.4 Technical validation

Quality assessment using a 5-fold cross-validation (WIENS et al., 2008) of the training samples indicates an overall accuracy of 0.96. Table 3.5 shows the user's and producer's accuracy for each land use and cover classes.

Table 3.5 - Summary of k-fold cross-validation accuracy estimation.

Labels	**User's accuracy**	**Producer's accuracy**
Cerrado	0.98	0.99
Fallow-Cotton	0.96	0.93
Forest	0.99	0.98
Pasture	0.97	0.98
Soy-Corn	0.91	0.93
Soy-Cotton	0.97	0.97
Soy-Fallow	0.98	0.98
Soy-Millet	0.90	0.89
Soy-Sunflower	0.77	0.65
Overall Accuracy: 0.96		

The cropland area increased from 2001 to 2017. This trend is corroborated by (SPERA et al., 2014; ARVOR et al., 2013; KASTENS et al., 2017). A decreasing on Soybean area between 2005 to 2007 was also observed by (SPERA et al., 2014; ARVOR et al., 2012). Moreover, the use of double-crop systems, involving soybeans in the first cycle and some other commercial crops in the second cycle, also increased from 2001 to 2017. This is in accordance with (KASTENS et al., 2017).

The correlation coefficient between the agricultural areas classified by our method and the official crop statistics by the Brazilian Institute of Geography and Statistics (IBGE) (INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA - IBGE, 2018), for harvests from 2001 to 2017, was equal to 0.98. At the state level, soybean, cotton, corn, and sunflower areas had a correlation equal to 0.97, 0.85, 0.98, and 0.80 respectively, as shown in Figure 3.5.

Compared to the IBGE statistics, the classification slightly underestimates soybean areas in most years. It also underestimates cotton areas until 2012 and overestimates corn areas. IBGE statistics are based on questionnaires and not on systematic surveys,

which might produce inaccurate estimates. Additionally, these differences may have been caused by the spatial resolution of MODIS (250 meters), which generates spectral mixing for different land uses within a single pixel (ZHONG et al., 2016).

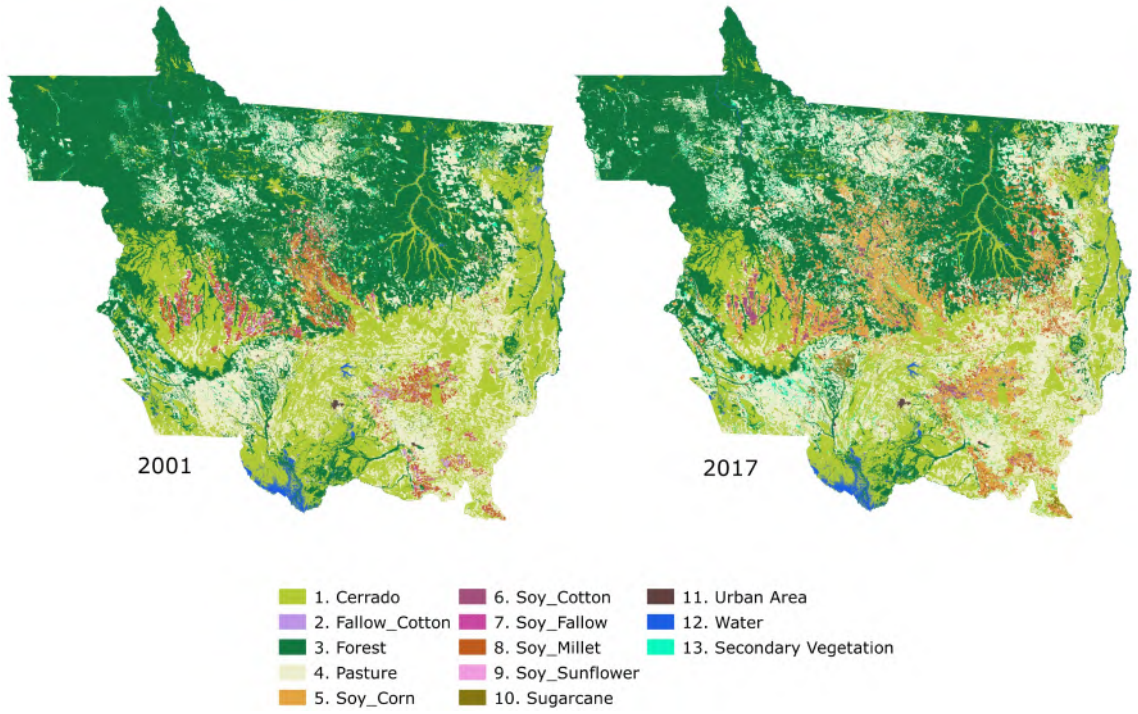
Forest area has a correlation of 0.88 if compared with data of PRODES (ALMEIDA et al., 2016). The classification method overestimated forest areas in Mato Grosso, possibly because some cerrado areas are very dense and spectrally similar to forest. We also compared the areas classified as forest with the Global Maps of 21st-Century Forest Cover Change produced by (HANSEN et al., 2013), for the year 2000. We found that 98% of the pixels classified as forest match the pixels above 25% of tree cover.

Our results show an expansion of pasture area in the Mato Grosso State between 2001 and 2017. We can observe in Figure 3.6 that pasture expansion occurred mainly in the north, within the Amazon biome. A similar finding is observed by (PARENTE; FERREIRA, 2018).

All these validation results show that our maps are consistent and reliable. However, the spatial resolution of MODIS images (250m) have limitations to represent some areas as small crop fields, water bodies, and urban areas.

Figure 3.5 - Comparison of land use areas from 2001 to 2017. Total area of a) soybean, b) corn, c) cotton, and d) forest in Mato Grosso estimated by the proposed classification method, together with IBGE cropland survey and PRODES.

Figure 3.6 - Classified maps of Mato Grosso in 2001 (left) and 2017 (right), with sugarcane, urban area and water masks.



3.5 Usage notes

The land use and cover data set contributes to understanding land use changes and trends. It provides essential information for public policy, decision making, environmental studies, territorial planning, and law and agreements enforcement. In this section, we provide two applications examples of land use and cover data set.

The data set can provide information to assist in the fulfilment of Brazilian NDC, which commit to reduce greenhouse gas emissions (GHG) by 37% below 2005 levels in 2025 and 43% until 2030. Actions comprising land use, renewable energy, and low carbon agriculture sectors compose key elements of the Brazilian commitments. Land use and cover maps are fundamental for monitoring land use change trends. These maps can also be used as input to complex models for GHG inventories to verify if the Brazilian NDC has been fulfilled for climate mitigation.

The demand for agricultural land is one of the main drivers of land use change in Brazil. There are already public political aggravations to slow down the expansion, especially of soybeans (whose Mato Grosso is the largest producer state in the world) and encourage the intensification of agriculture, as for example soybean moratorium

(signed in 2006) and the new Forest Code (NFC) (signed in 2012). Another potential usage of the data set is to track the agriculture intensification in Brazil, and on which land use agriculture has expanded. It is also possible to verify compliance with the soybean moratorium and the NFC policies.

4 RSTAC: AN R PACKAGE TO ACCESS SPATIOTEMPORAL ASSET CATALOG SATELLITE IMAGERY

This Chapter[1] presents the *rstac* an *R* package that implements a client for the SpatioTemporal Assets Catalog (STAC) API, a simple and extensible document-based easy-to-read specification adopted by important players of Earth observation data providers. STAC API enables query and access to a growing set of global satellite imagery providers. The adoption of common protocols is critical to interoperability of systems in an heterogeneous cloud computing environments and allows automated access to petabytes of images of a several Earth observation satellites. Here, we demonstrate how *rstac* can be used in an application of land use and land cover mapping.

4.1 Introduction

Only in 2019, 5 PB of terrestrial surface images were produced by Landsat, MODIS, and Sentinel series satellites (SOILLE et al., 2018). The majority of these images were stored in cloud computing platforms by most space agencies that have adopted an open data policy. Cloud computing are changing how all these big Earth Observation data is being distributed. One of the main challenges is to organise, access, and process these huge data sets in an automatic and interoperable way (GIULIANI et al., 2019).

A key piece to solving part of this challenge are protocol standards to catalogue, search, and access these data sets. The STAC is a simple and extensible specification of JSON documents based on OGC OAFeat core (OPEN GEOSPATIAL CONSORTIUM - OCG, 2019) that is being well accepted by important EO data providers like USGS, AWS, and Microsoft. The *rstac* is the first package related to the STAC API specification written in the R language. It intends to facilitate the usage of STAC API technology by the *R* community.

This Chapter aims to present the *rstac* package, describing its software design and demonstrate one application. The document is organised as follows: first, we present the context of the STAC specification and its main elements; second, we describe

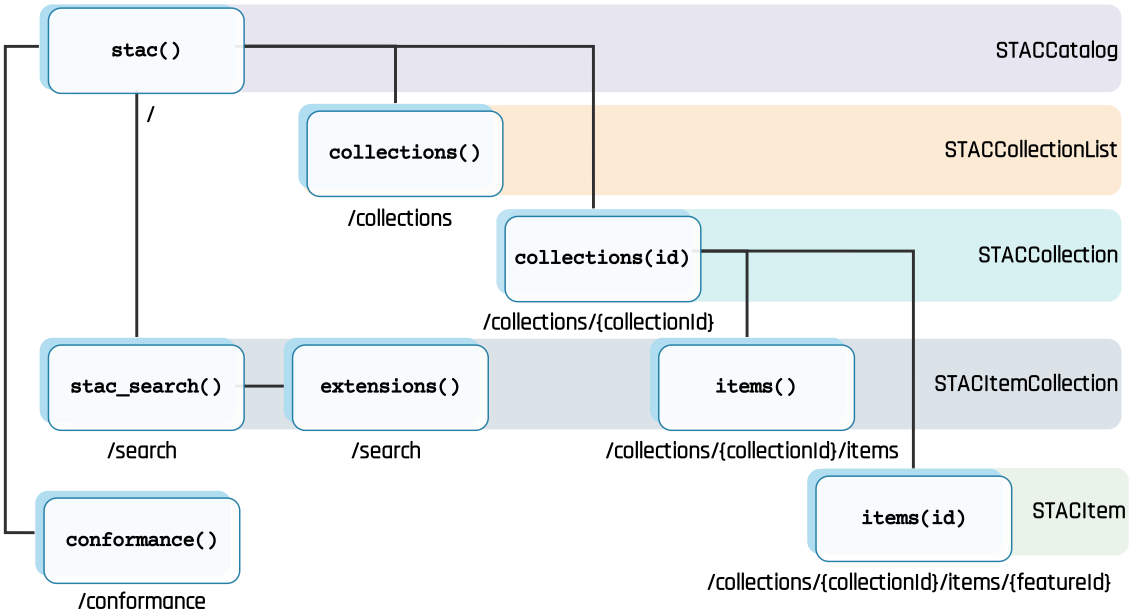
[1] A version of this Chapter was accepted as a conference paper in the 2021 International Geoscience and Remote Sensing Symposium (IGARSS 2021), and was co-authored by Felipe Souza, Matheus Zaglia, Gilberto Queiroz, Rafael Santos, Karine Ferreira.

how the package implements the access to the STAC API; finally, we show how the technology can be used in an EO data cube context by describing an example of land use and land cover classification.

4.2 The STAC specification

Often, the adoption of complex specifications can hinder the development of applications. This is the case for some of the traditional OGC Web services like WFS 2 (OPEN GEOSPATIAL CONSORTIUM - OCG, 2014). To keep things running at a minimum, developers need to read hundreds of specification pages. One of the main features of the STAC specification is its simplicity and extensibility. The standard defines a basic core structure in JSON (BRAY, 2017) and GeoJSON (BUTLER et al., 2016) that can be easily extended. It follows the recommendations of good practices for web spatial data (TANDY et al., 2017), which includes the adoption of a metadata format that can be easy to be read by humans, the description of satellite images and web links to access them, the possibility to browsing the catalogue entirely by following web links so that static catalogues can be defined without the need for an API service, and a search API to retrieve the same documents dynamically. These principles can be observed in the two components of the specification, the STAC documents and the STAC API.

Figure 4.1 - The hierarchical structure of the *rstac* package, composed of functions (boxes), endpoints (text below boxes), and classes (horizontal bars).



The documents "STAC Catalogs" and "STAC Collections" are regular JSON files used to organise the catalogue hierarchically and provide some general information of the collections. They help split many catalogue items into small groups to facilitate users to browse into extensive static catalogues. The documents "STAC Item Collection" and "STAC Item" are extended GeoJSON files that describe and provide links to access spatiotemporal assets. The former lists a set of items matched by a query, and the latter represents a single item, the core unit of the catalogue. Besides the static catalogues, STAC documents can be dynamically obtained by the search API. The API is defined in OpenAPI 3 (OPEN API INITIATIVE - OAI, 2020) and adheres to the core principles of RESTful applications (TANDY et al., 2017).

The API consists of URL paths known as endpoints, to which HTTP requests are made, and JSON documents returned according to provided query parameters. STAC API is a superset of OGC API Features (OAFeat) (OPEN GEOSPATIAL CONSORTIUM - OCG, 2019) specification and hence, adopts all its endpoints. It also extends the OAFeat by proposing a new endpoint, `/search`. This endpoint enables the searching for items across multiple collections and makes complex queries to filter items to be returned by the service. Different from static catalogues, users can browse the matched items through a pagination mechanism.

4.3 Software design

The *rstac* package was developed entirely in *R* , following the principles discussed in (WICKHAM, 2019b). The document specifies four principles: the re-usability of native R functions, communication between operations, usage functions involving functional programming, and design for the general public. The first one is implemented by taking advantage of the S3 object model and extending base R functions like `print()`. Also, all *rstac* documents are S3 subclass of ordinary lists, and all functions developed for this type can be applied to them either. The remaining three principles define the design of *rstac* , which can be divided into three main components: *query*, *request*, and *documents*. The query component consists of building the set of parameters passed to the STAC service through an HTTP request. The next component, request, defines the HTTP protocol parameters, i.e., HTTP verb, headers, and content-encoding. Some of the STAC API behavior depends on these parameters. Finally, the document component starts when a response is retrieved and parsed. The document can then be consumed, items browsed, and assets downloaded.

The query construction in *rstac* is based on a hierarchy of endpoint function calls. To simplify its creation and make the process user-friendly, the functions calling order resembles the endpoints paths. Figure 4.1 shows a tree denoting the order in which query functions (boxes) can be called and the resulting *rstac* object representing the document returned by the server (horizontal bars). The corresponding endpoints are shown below in the functions boxes. Every new query starts from `stac()` endpoint function (at the top of the hierarchy) and, using a *Unix*-like pipe operator (`%>%` in *R*), one can make successive calls to other endpoint functions to form the desired query.

Besides providing full support for all endpoints from the STAC specification, *rstac* offers extra functions that can be used on documents returned from a request. For example, for *STAC Item Collection* documents, it is possible to check the object's length and how many items were returned according to the search criteria, using the functions `items_length()` and `items_matched()`, respectively. Since some servers have a maximum limit of items to be returned per search, the *rstac* package offers a paging function that returns all search items, using the `items_fetch()` function. Furthermore, the user can download all the search assets through the `assets_download()` function.

All these components are put together by a command chain designed to be used in a functional programming way: the query can be progressively built before the request by calling *rstac* endpoint functions; the request concerns only with service communication, such as authentication and content-encoding; the resulting document passed to further operations. An example of this use is illustrated in the Listing 4.1.

Listing 4.1 - Extracting time series from a a data cube.

```
# BDC stac service
stac("http://brazildatacube.dpi.inpe.br/stac") %>%
    collections("CB4_64_16D_STK-1") %>% # cube CBERS
    items(bbox = c(-45.661502, -12.534252, -45.426156, -12.390242)) %>%
        # delimit extent
    items(datetime = "2018-09-01/2019-08-28") %>% # delimit date range
    get_request() %>% # do request
    items_fetch() # retrieve all matched entries
```

This design based on the *R* S3 object model enables *rstac* to be extended at different levels, either by adding new endpoints, new query parameters, or new document operations. These extensions facilitate the package to be up-to-date according to

specification developments and allow the implementation of new applications on it. The *rstac* provides a set of functions and complete documentation to help the development of new extensions.

4.4 Application

To demonstrate an application of STAC technology, we describe an example of land use and land change classification that uses machine learning and satellite image time series. The time series data were retrieved from the Brazil Data Cube (BDC) project (FERREIRA et al., 2020a). The BDC project aims to create multidimensional data cubes ready for analysis from remote sensing images of medium resolution (10 to 64-meters) of CBERS, Landsat, and Sentinel satellites series, for the entire Brazilian territory. One of these data cubes' main applications in the project context is the generation of thematic maps of land use and land cover classifications from Brazilian biomes (FERREIRA et al., 2020a).

4.4.1 Land use and land cover classification of satellite image time series

The classification is performed by the *sits* [2] *R* package. Once we have trained a machine learning classifier, the mapping is done by reading multiple images from an area of interest to reconstruct the time series and pass them to the classifier, which gives back the land use and land cover classes for each time series. Figure 4.2 illustrate how the integration between *sits* and *rstac* works.

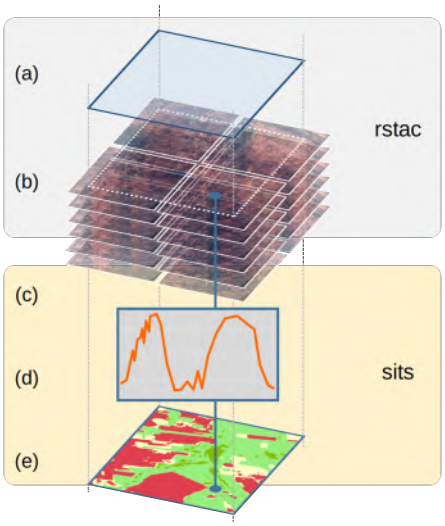
The process starts by defining an area and an interval of interest. This area can be represented by a bounding box and the interval by two dates queried by *rstac* to the BDC STAC server. The metadata returned is used by *sits* to retrieve the time series for each pixel classified and then generates a land use and land cover map. Here, we defined our region of interest inside the Bahia State, Brazil (see Figure 4.3-(c)). The region has approximately 40,000 ha and is bounded by the coordinates (long, lat) $-45.7564, -12.6708$, and $-45.3207, -12.2626$ (in the EPSG:4326 reference system).

We used 922 samples from the same region. These samples were collected by experts using high-resolution images and were provided by the BDC project. The samples comprise four land use and land cover classes: "Crops", "Shrublands", "Other vegetation", and "Pasture". We used *rstac* to extract the Normalized Difference Vegetation Index (NDVI) time series of each sample from CBERS-4 satellite data cube. This

[2] Note: package discussed in Chapter 2

EO data cube is formed by 16-days of composite images from the AWFI sensor with 64-meters of spatial resolution. We defined the time series interval from 2018-09-01 to 2019-08-28, which matches the Brazilian annual crop calendar.

Figure 4.2 - *rstac* and *sits* integration. Data flows downward: (a) region and interval of interest; (b) STAC query; (c) time series retrieving; (d) machine learning classification; and (e) land use and land cover map.

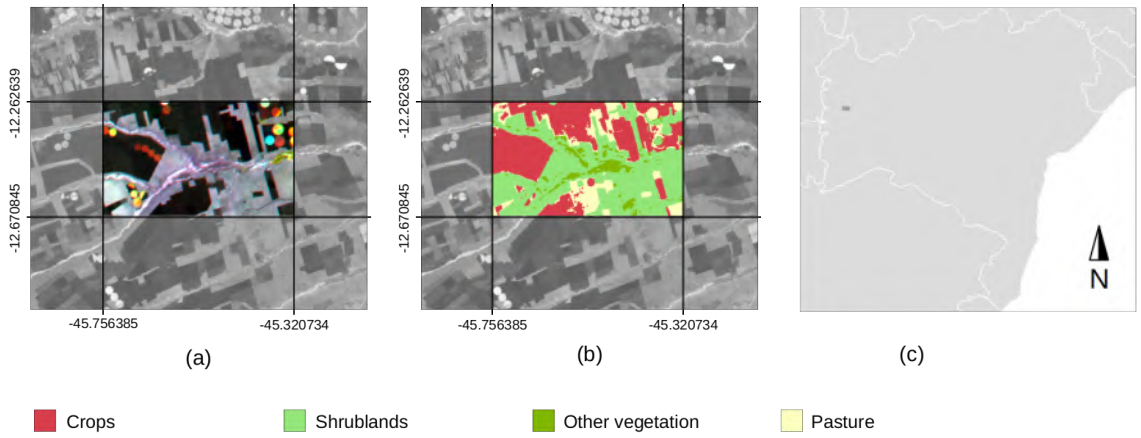


We trained a 1000-trees Random Forest (BREIMAN, 2001) model using *sits* . The trained model produced a one-year thematic map that is shown in Figure 4.3-(b). The full methodology is described in (SIMOES et al., 2020). To accomplish this classification, we accessed 24 NDVI images that formed all the observations for the period. All-time series were extracted using GDAL virtual file system driver by HTTP requests. The classification process took less than a minute.

4.5 Conclusions

One of the main challenges for EO technology developers is the interoperability. To achieve an automation level capable to query and access a variety of repositories located in different platforms, an abstraction layer of protocols must be adopted. STAC is becoming an feasible candidate to support the interoperability required to access EO data cubes. This is even more critical considering the trends of EO community to use cloud computing from different vendors or public organisations.

Figure 4.3 - Land use and land cover classification using rstac and sits: (a) false-color composition formed by NDVI on different dates (R: 2018-09-14, G: 2018-09-30, B: 2018-10-16); (b) land use and land cover map; (c) region (little dark gray rectangle) located in western Bahia State, Brazil.



Based on a REST+JSON solution, the STAC provides only basic features but is easy to extend. Following the same principle, the *rstac* package is designed to be extensible at many levels of specification, facilitating the development of new applications based on the protocol. The *rstac* was the first implementation concerning STAC in the R language, which can leverage STAC-based technologies in the *R* community, such as *sits* , that have been enabled to access different EO data repositories, which facilitates the process of environment preparation in any cloud computing platform.

Here, we provided one little example that shows how STAC protocol can be used to integrate software and EO data cubes. We presented a land use and land cover classification using time series by accessing the CBERS-4 data cube from the BDC cloud environment. All processes involved the integration between *rstac* and *sits* in the client side and STAC Web service in the server side.

5 FINAL REMARKS

Developers of analytical software for big EO data face several challenges. The *sits* software makes the hypothesis that image time series provide an adequate description of land use and land cover change. Instead of using phenological or temporal metrics, we use machine learning to find the patterns that discriminate the labels. Our results on generating good quality maps are supportive of this hypothesis.

Our method faces challenges because of the massive use of data. The parallel processing design of the package considered this issue. It is scalable and exploits the available computational resources, achieving good computational performance.

The design of *sits* prioritised the support to the different computing environments such as AWS and Azure. In our vision, the emergent scenario in the EO community is that there will be multiple data cube providers making use of different cloud computing services. This heterogeneity of data and computational environment imposes interoperability challenges if we want the same software to work in different cloud services with similar results. Another relevant decision was to use the STAC protocol to describe the contents of image cloud services.

Future steps for *sits* include supporting to new machine learning classifiers, optimising the access to EO data cubes over ARD image collections, including arithmetic over data cubes. One important improvement is in the training phase, by supporting techniques such as active learning and semi-supervised self learning. We also intend to include into *sits* multiyear classification post-processing by native *sits* instructions. Given the global applicability of *sits* , we intend to support a user and developer community by providing guidance and documentation.

REFERENCES

ADAMI, M.; RUDORFF, B. F. T.; FREITAS, R. M.; AGUIAR, D. A.; SUGAWARA, L. M.; MELLO, M. P. Remote sensing time series to evaluate direct land use change of recent expanded sugarcane crop in Brazil. **Sustainability**, v. 4, n. 4, p. 574–585, 2012. 35

ALLEN, J.; FERGUSON, G. Actions and events in interval temporal logic. **Journal of Logic and Computation**, v. 4, n. 5, p. 531–579, oct. 1994. ISSN 0955-792X. 6

ALMEIDA, C.; COUTINHO, A.; ESQUERDO, J.; ADAMI, M.; VENTURIERI, A.; DINIZ, C.; DESSAY, N.; DURIEUX, L.; GOMES, A. High spatial resolution land use and land cover mapping of the Brazilian Legal Amazon in 2008 using Landsat-5/TM and MODIS data. **Acta Amazonica**, v. 46, n. 3, p. 291–302, sep. 2016. 39

AMANI, M.; GHORBANIAN, A.; AHMADI, S. A.; KAKOOEI, M.; MOGHIMI, A.; MIRMAZLOUMI, S. M.; MOGHADDAM, S. H. A.; MAHDAVI, S.; GHAHREMANLOO, M.; PARSIAN, S.; WU, Q.; BRISCO, B. Google Earth Engine cloud computing platform for remote sensing big data applications: a comprehensive review. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, v. 13, p. 5326–5350, 2020. ISSN 2151-1535. 1, 24

APPEL, M.; PEBESMA, E. On-demand processing of data cubes from satellite image collections with the gdalcubes library. **Data**, v. 4, n. 3, p. 1–16, 2019. 5, 7, 10

ARÉVALO, P.; BULLOCK, E. L.; WOODCOCK, C. E.; OLOFSSON, P. A Suite of tools for continuous land change monitoring in Google Earth Engine. **Frontiers in Climate**, v. 2, 2020. ISSN 2624-9553. 4

ARNETH, A.; SITCH, S.; PONGRATZ, J.; STOCKER, B. D.; CIAIS, P.; POULTER, B.; BAYER, A. D.; BONDEAU, A.; CALLE, L.; CHINI, L. P.; GASSER, T.; FADER, M.; FRIEDLINGSTEIN, P.; KATO, E.; LI, W.; LINDESKOG, M.; NABEL, J. E. M. S.; PUGH, T. A. M.; ROBERTSON, E.; VIOVY, N.; YUE, C.; ZAEHLE, S. Historical carbon dioxide emissions caused by land-use changes are possibly larger than assumed. **Nature Geoscience**, v. 10, n. 2, p. 79–84, feb. 2017. ISSN 1752-0894, 1752-0908. 1

ARVOR, D.; DURIEUX, L.; ANDRÉS, S.; LAPORTE, M.-A. Advances in geographic object-based image analysis with ontologies: a review of main contributions and limitations from a remote sensing perspective. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 82, p. 125–137, aug. 2013. ISSN 0924-2716. 38

ARVOR, D.; JONATHAN, M.; MEIRELLES, M. S. P.; DUBREUIL, V.; DURIEUX, L. Classification of MODIS EVI time series for crop mapping in the state of Mato Grosso, Brazil. **International Journal of Remote Sensing**, v. 32, n. 22, p. 7847–7871, nov. 2011. ISSN 0143-1161, 1366-5901. 4, 28, 30

ARVOR, D.; MEIRELLES, M.; DUBREUIL, V.; SHIMABUKURO, Y. E. Analyzing the agricultural transition in Mato Grosso, Brazil, using satellite-derived indices. **Applied Geography**, v. 32, n. 2, p. 702–713, 2012. 4, 38

BELGIU, M.; DRAGUT, L. Random forest in remote sensing: a review of applications and future directions. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 114, p. 24–31, 2016. 14

BLOCH, J. How to design a good API and why it matters. In: ACM SIGPLAN SYMPOSIUM ON OBJECT-ORIENTED PROGRAMMING SYSTEMS, LANGUAGES AND APPLICATIONS, 21., 2006. **Proceedings...** New York: ACM, 2006. p. 506–507. ISBN 978-1-59593-491-8. 9

BRAY, T. The JavaScript Object Notation (JSON) data interchange format. **Internet Engineering Task Force (IETF)**, RFC 8259, 2017. 44

BREIMAN, L. Random forests. **Machine Learning**, v. 45, n. 1, p. 5–32, 2001. ISSN 08856125. 48

BUTLER, H.; DALY, M.; DOYLE, A.; GILLIES, S.; HAGEN, S.; SCHAUB, T. The GeoJSON format. **Internet Engineering Task Force (IETF)**, RFC 7946, 2016. 44

CAMARA, G.; ASSIS, L. F.; RIBEIRO, G.; FERREIRA, K. R.; LLAPA, E.; VINHAS, L. Big Earth observation data analytics: matching requirements to system architectures. In: ACM SIGSPATIAL INTERANTIONAL WORKSHOP ON ANALYTICS FOR BIG GEOSPATIAL DATA, 5., 2016. **Proceedings...** New York: ACM, 2016. p. 1–6. 1, 5

CAMARA, G.; EGENHOFER, M. J.; FERREIRA, K.; ANDRADE, P.; QUEIROZ, G.; SANCHEZ, A.; JONES, J.; VINHAS, L. Fields as a generic data type for big spatial data. In: DUCKHAM, M.; PEBESMA, E.; STEWART, K.; FRANK, A. U. (Ed.). **Geographic information science**. [S.l.]: Springer, Cham, 2014. v. 8728, p. 159–172. 5

CAMARA, G.; PICOLI, M.; SIMOES, R.; MACIEL, A.; CARVALHO, A.; COUTINHO, A.; ESQUERDO, J.; ANTUNES, J.; BEGOTTI, R.; ARVOR, D. **Land cover change maps for Mato Grosso State in Brazil: 2001-2016**. PANGAEA, 2019. Available from: <https://doi.org/10.1594/PANGAEA.899706>. Access in: 11 May 2020. 30, 37

CHEN, T.; GUESTRIN, C. XGBoost: a scalable tree boosting system. In: ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 22., 2016, New York. **Proceedings...** New York: ACM, 2016. p. 785–794. ISBN 978-1-4503-4232-2. 14

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, v. 20, n. 3, p. 273–297, 1995. 33

CRESSIE, N. Bayesian smoothing of rates in small geographic areas. **Journal of Regional Science**, v. 35, n. 4, p. 659–673, 1995. ISSN 1467-9787. 34

DEAN, J.; GHEMAWAT, S. MapReduce: simplified data processing on large clusters. **Communications of the ACM**, v. 51, n. 1, p. 107–113, jan. 2008. ISSN 0001-0782, 1557-7317. 24

DEL-CLARO, K.; TOREZAN-SILINGARDI, H. M. The study of biotic interactions in the Brazilian Cerrado as a path to the conservation of biodiversity. **Anais da Academia Brasileira de Ciências**, v. 91, n. suppl 3, p. e20180768, 2019. ISSN 1678-2690, 0001-3765. 19

DIDAN, K. **MOD13Q1 MODIS/Terra Vegetation Indices 16-Day L3 Global 250m SIN Grid V006**. 2015. Available from: <http://doi.org/10.5067/MODIS/MOD13Q1.006>. Access in: 11 May 2018. 30

FAWAZ, H.; LUCAS, B.; FORESTIER, G.; PELLETIER, C.; SCHMIDT, D. F.; WEBER, J.; WEBB, G. I.; IDOUMGHAR, L.; MULLER, P.-A.; PETITJEAN, F. InceptionTime: finding AlexNet for time series classification. **Data Mining and Knowledge Discovery**, v. 34, n. 6, p. 1936–1962, nov. 2020. ISSN 1573-756X. 14

FAWAZ, H. I.; FORESTIER, G.; WEBER, J.; IDOUMGHAR, L.; MULLER, P.-A. Deep learning for time series classification: a review. **Data Mining and Knowledge Discovery**, v. 33, n. 4, p. 917–963, 2019. 5, 14

FERREIRA, K.; QUEIROZ, G.; CAMARA, G.; SOUZA, R.; VINHAS, L.; MARUJO, R.; SIMOES, R.; NORONHA, C.; COSTA, R.; ARCANJO, J.; GOMES, V.; ZAGLIA, M. Using remote sensing images and cloud services on AWS to improve land use and cover monitoring. In: LATIN AMERICAN GRSS & ISPRS REMOTE SENSING CONFERENCE, 2020, Santiago, Chile. **Proceedings...** . Santiago: GRSS/ISPRS, 2020. 1, 5, 7, 8, 14, 20, 47

FERREIRA, K. R.; QUEIROZ, G. R.; VINHAS, L.; MARUJO, R. F. B.; SIMOES, R. E. O.; PICOLI, M. C. A.; CAMARA, G.; CARTAXO, R.; GOMES, V. C. F.; SANTOS, L. A.; SANCHEZ, A. H.; ARCANJO, J. S.; FRONZA, J. G.; NORONHA, C. A.; COSTA, R. W.; ZAGLIA, M. C.; ZIOTI, F.; KORTING, T. S.; SOARES, A. R.; CHAVES, M. E. D.; FONSECA, L. M. G. Earth observation data cubes for Brazil: requirements, methodology and products. **Remote Sensing**, v. 12, n. 24, p. 4033, jan. 2020. 20

FLYNN, D. F. B.; GOGOL-PROKURAT, M.; NOGEIRE, T.; MOLINARI, N.; RICHERS, B. T.; LIN, B. B.; SIMPSON, N.; MAYFIELD, M. M.; DECLERCK, F. Loss of functional diversity under land use intensification across multiple taxa. **Ecology Letters**, v. 12, n. 1, p. 22–33, jan. 2009. ISSN 1461023X, 14610248. 1

FOLEY, J. A.; DEFRIES, R.; ASNER, G. P.; BARFORD, C.; BONAN, G.; CARPENTER, S. R.; CHAPIN, F. S.; COE, M. T.; DAILY, G. C.; GIBBS, H. K.; HELKOWSKI, J. H.; HOLLOWAY, T.; HOWARD, E. A.; KUCHARIK, C. J.; MONFREDA, C.; PATZ, J. A.; PRENTICE, I. C.; RAMANKUTTY, N.; SNYDER, P. K. Global consequences of land use. **Science**, v. 309, n. 5734, p. 570–574, 2005. 1, 3

FRENAY, B.; VERLEYSEN, M. Classification in the presence of label noise: a survey. **IEEE Transactions on Neural Networks and Learning Systems**, v. 25, n. 5, p. 845–869, may 2014. ISSN 2162-2388. 12, 31

GALFORD, G. L.; MUSTARD, J. F.; MELILLO, J.; GENDRIN, A.; CERRI, C. C.; CERRI, C. E. Wavelet analysis of MODIS time series to detect expansion and intensification of row-crop agriculture in Brazil. **Remote Sensing of Environment**, v. 112, n. 2, p. 576–587, 2008. 4

GALTON, A. Fields and objects in space, time, and space-time. **Spatial Cognition & Computation**, v. 4, n. 1, p. 39–68, mar. 2004. ISSN 1387-5868. 5

GARNOT, V. S. F.; LANDRIEU, L. Lightweight temporal self-attention for classifying satellite image time series. **arXiv:2007.00586** [**cs**], jul. 2020. 4, 14

GHEMAWAT, S.; GOBIOFF, H.; LEUNG, S.-T. The Google file system. In: ACM SYMPOSIUM ON OPERATING SYSTEMS PRINCIPLES, 19., 2003, New York. **Proceedings...** New York: ACM, 2003. p. 29–43. ISBN 978-1-58113-757-6. 24

GHIMIRE, B.; ROGAN, J.; MILLER, J. Contextual land-cover classification: incorporating spatial dependence in land-cover classification models using random forests and the Getis statistic. **Remote Sensing Letters**, v. 1, n. 1, p. 45–54, 2010. 34

GIBBS, H. K.; RAUSCH, L.; MUNGER, J.; SCHELLY, I.; MORTON, D. C.; NOOJIPADY, P.; SOARES-FILHO, B.; BARRETO, P.; MICOL, L.; WALKER, N. F. Brazil's soy moratorium. **Science**, v. 347, n. 6220, p. 377–378, jan. 2015. ISSN 0036-8075, 1095-9203. 28

GIULIANI, G.; CAMARA, G.; KILLOUGH, B.; MINCHIN, S. Earth observation open science: enhancing reproducible science using data cubes. **Data**, v. 4, n. 4, p. 147, dec. 2019. 4, 43

GIULIANI, G.; CHATENOUX, B.; PILLER, T.; MOSER, F.; LACROIX, P. Data Cube on Demand (DCoD): generating an Earth observation data cube anywhere in the world. **International Journal of Applied Earth Observation and Geoinformation**, v. 87, p. 102035, may 2020. ISSN 0303-2434. 5, 24

GOMES, V. C. F.; QUEIROZ, G. R.; FERREIRA, K. R. An overview of platforms for big Earth observation data management and analysis. **Remote Sensing**, v. 12, n. 8, p. 1253, jan. 2020. 1

GONG, P.; HOWARTH, P. J. Performance analyses of probabilistic relaxation methods for land-cover classification. **Remote Sensing of Environment**, v. 30, n. 1, p. 33–42, 1989. ISSN 0034-4257. 34

GOODLAND, R. A Physiognomic analysis of the 'Cerrado' vegetation of central Brasil. **The Journal of Ecology**, v. 59, n. 2, p. 411, jul. 1971. ISSN 00220477. 19

GORELICK, N.; HANCHER, M.; DIXON, M.; ILYUSHCHENKO, S.; THAU, D.; MOORE, R. Google Earth Engine: planetary-scale geospatial analysis for everyone. **Remote Sensing of Environment**, v. 202, p. 18–27, 2017. 4, 24

HANSEN, M. C.; POTAPOV, P. V.; MOORE, R.; HANCHER, M.; TURUBANOVA, S. A.; TYUKAVINA, A.; THAU, D.; STEHMAN, S. V.; GOETZ, S. J.; LOVELAND, T. R.; KOMMAREDDY, A.; EGOROV, A.; CHINI, L.; JUSTICE, C. O.; TOWNSHEND, J. R. G. High-resolution global maps of 21st-century forest cover change. **Science**, v. 342, n. 6160, p. 850–853, 2013. 39

HANSON, M. The Open-source software ecosystem for leveraging public datasets in Spatio-Temporal Asset Catalogs (STAC). **AGU Fall Meeting Abstracts**, v. 23, dec. 2019. 2, 10

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The elements of statistical learning: data mining, inference, and prediction**. New York: Springer, 2009. 34

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA - IBGE. **Agricultural census**. Rio de Janeiro: IBGE, 2018. 38

_____. **Monitoramento da cobertura e uso da terra do Brasil: 2016–2018**. Rio de Janeiro: IBGE, 2020. 26 p. 20

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS - INPE. **Amazon deforestation monitoring project (PRODES)**. São José dos Campos, SP: INPE, 2019. 35

_____. **PRODES - incremento anual de area desmatada no Cerrado brasileiro**. São José dos Campos, SP: INPE, 2019. 35

JONSSON, P.; EKLUNDH, L. TIMESAT: a program for analyzing time-series of satellite sensor data. **Computers and Geosciences**, v. 30, n. 8, p. 833–845, 2004. ISSN 0098-3004. 4

KASTENS, J.; BROWN, J.; COUTINHO, A.; BISHOP, C.; ESQUERDO, J. Soy moratorium impacts on soybean and deforestation dynamics in Mato Grosso, Brazil. **PLOS ONE**, v. 12, n. 4, p. e0176168, 2017. 28, 30, 38

KENNEDY, R. E.; YANG, Z.; COHEN, W. B. Detecting trends in forest disturbance and recovery using yearly Landsat time series: 1. LandTrendr — temporal segmentation algorithms. **Remote Sensing of Environment**, v. 114, n. 12, p. 2897–2910, dec. 2010. ISSN 00344257. 4

KOHONEN, T. The self-organizing map. **Proceedings of the IEEE**, v. 78, n. 9, p. 1464–1480, sep. 1990. ISSN 1558-2256. 12, 31

KUENZER, C.; DECH, S.; WAGNER, W. **Remote sensing time series revealing land surface dynamics: status quo and the pathway ahead**. [S.l.]: Springer, 2015. ISBN 978-3-319-15967-6. 1

LAMBIN, E. F.; GEIST, H. J.; LEPERS, E. Dynamics of land-use and land-cover change in tropical regions. **Annual Review of Environment and Resources**, v. 28, n. 1, p. 205–241, 2003. 4

LEWIS, A.; OLIVER, S.; LYMBURNER, L.; EVANS, B.; WYBORN, L.; MUELLER, N.; RAEVKSI, G.; HOOKE, J.; WOODCOCK, R.; SIXSMITH, J.; WU, W.; TAN, P.; LI, F.; KILLOUGH, B.; MINCHIN, S.; ROBERTS, D.; AYERS, D.; BALA, B.; DWYER, J.; DEKKER, A.; DHU, T.; HICKS, A.; IP, A.; PURSS, M.; RICHARDS, C.; SAGAR, S.; TRENHAM, C.; WANG, P.; WANG, L.-W. The Australian Geoscience Data Cube — foundations and lessons learned. **Remote Sensing of Environment**, v. 202, p. 276–292, dec. 2017. ISSN 00344257. 5, 24

MACIEL, A.; CAMARA, G.; VINHAS, L.; PICOLI, M.; BEGOTTI, R.; ASSIS, L. F. A spatiotemporal calculus for reasoning about land-use trajectories. **International Journal of Geographical Information Science**, v. 33, n. 1, p. 176–192, 2019. 36

MARTINELLI, L. A.; NAYLOR, R.; VITOUSEK, P. M.; MOUTINHO, P. Agriculture in Brazil: impacts, costs, and opportunities for a sustainable future. **Current Opinion in Environmental Sustainability**, v. 2, n. 5, p. 431–438, 2010. ISSN 1877-3435. 27

MAUS, V.; CÂMARA, G.; APPEL, M.; PEBESMA, E. dtwSat: time-weighted dynamic time warping for satellite image time series analysis in R. **Journal of Statistical Software**, v. 88, n. 5, p. 1–31, 2019. 4

MAUS, V.; CAMARA, G.; CARTAXO, R.; SANCHEZ, A.; RAMOS, F. M.; QUEIROZ, G. R. A time-weighted dynamic time warping method for land-use and land-cover mapping. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, v. 9, n. 8, p. 3729–3739, 2016. 4, 31

MAXWELL, A. E.; WARNER, T. A.; FANG, F. Implementation of machine-learning classification in remote sensing: an applied review. **International Journal of Remote Sensing**, v. 39, n. 9, p. 2784–2817, 2018. 12, 34

MOUNTRAKIS, G.; IM, J.; OGOLE, C. Support vector machines in remote sensing: a review. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 66, n. 3, p. 247–259, 2011. 14, 33

NOI, P. T.; KAPPAS, M. Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using Sentinel-2 imagery. **Sensors**, v. 18, n. 1, p. 18, jan. 2018. 12

OLOFSSON, P. **Understanding accuracy and area estimation**. Boston: Boston University, 2014. 18

OLOFSSON, P.; FOODY, G. M.; STEHMAN, S. V.; WOODCOCK, C. E. Making better use of accuracy data in land change studies: estimating accuracy and area and quantifying uncertainty using stratified estimation. **Remote Sensing of Environment**, v. 129, p. 122–131, feb. 2013. ISSN 0034-4257. 18, 23

OPEN API INITIATIVE - OAI. **OpenAPI specification 3.0.3**. 2020. Available from: <https://swagger.io/specification/>. Access in: 26 Sep 2020. 45

OPEN GEOSPATIAL CONSORTIUM - OCG. **OGC web feature service 2.0 interface standard - with corrigendum**. 2014. Available from: <http://docs.opengeospatial.org/is/09-025r2/09-025r2.html>. Access in: 11 May 2020. 44

_____. **OGC API - features - part 1: core**. 2019. Available from: <http://docs.opengeospatial.org/is/09-025r2/09-025r2.html>. Access in: 11 May 2020. 43, 45

PARENTE, L.; FERREIRA, L. Assessing the spatial and occupation dynamics of the Brazilian pasturelands based on the automated classification of MODIS images from 2000 to 2016. **Remote Sensing**, v. 10, n. 4, p. 606, apr. 2018. 39

PARENTE, L.; MESQUITA, V.; MIZIARA, F.; BAUMANN, L.; FERREIRA, L. Assessing the pasturelands and livestock dynamics in Brazil, from 1985 to 2017: a novel approach based on high spatial resolution imagery and Google Earth Engine cloud computing. **Remote Sensing of Environment**, v. 232, p. 111301, oct. 2019. ISSN 0034-4257. 14

PARENTE, L.; NOGUEIRA, S.; BAUMANN, L.; ALMEIDA, C.; MAURANO, L.; AFFONSO, A. G.; FERREIRA, L. Quality assessment of the PRODES Cerrado deforestation data. **Remote Sensing Applications: Society and Environment**, v. 21, p. 100444, jan. 2021. ISSN 2352-9385. 8

PARENTE, L.; TAQUARY, E.; SILVA, A. P.; SOUZA, C.; FERREIRA, L. Next generation mapping: combining deep learning, cloud computing, and big remote sensing data. **Remote Sensing**, v. 11, n. 23, p. 2881, jan. 2019. 1, 20

PASQUARELLA, V. J.; HOLDEN, C. E.; KAUFMAN, L.; WOODCOCK, C. E. From imagery to ecology: leveraging time series of all available Landsat observations to map and monitor ecosystem state and dynamics. **Remote Sensing in Ecology and Conservation**, v. 2, n. 3, p. 152–170, 2016. ISSN 2056-3485. 4

PEBESMA, E. Simple features for R: standardized support for spatial vector data. **The R Journal**, v. 10, n. 1, p. 439, 2018. ISSN 2073-4859. 11

PEKEL, J. F.; COTTAM, A.; GORELICK, N.; BELWARD, A. S. High-resolution mapping of global surface water and its long-term changes. **Nature**, v. 540, p. 418–422, 2016. 35

PELLETIER, C.; VALERO, S.; INGLADA, J.; CHAMPION, N.; DEDIEU, G. Assessing the robustness of Random Forests to map land cover with high resolution satellite image time series over large areas. **Remote Sensing of Environment**, v. 187, p. 156–168, 2016. 4

PELLETIER, C.; WEBB, G. I.; PETITJEAN, F. Temporal convolutional neural network for the classification of satellite image time series. **Remote Sensing**, v. 11, n. 5, 2019. 4, 5, 14, 20

PETITJEAN, F.; KURTZ, C.; PASSAT, N.; GANÇARSKI, P. Spatio-temporal reasoning for the classification of satellite image time series. **Pattern Recognition Letters**, v. 33, n. 13, p. 1805–1815, oct. 2012. ISSN 0167-8655. 4

PICOLI, M.; CAMARA, G.; SANCHES, I.; SIMOES, R.; CARVALHO, A.; MACIEL, A.; COUTINHO, A.; ESQUERDO, J.; ANTUNES, J.; BEGOTTI, R. A.; ARVOR, D.; ALMEIDA, C. Big earth observation time series analysis for monitoring Brazilian agriculture. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 145, p. 328–339, 2018. 14, 28, 32, 34

PICOLI, M. C. A.; RORATO, A.; LEITÃO, P.; CAMARA, G.; MACIEL, A.; HOSTERT, P.; SANCHES, I. D. Impacts of public and private sector policies on soybean and pasture expansion in Mato Grosso–Brazil from 2001 to 2017. **Land**, v. 9, n. 1, p. 20, jan. 2020. 14

RUDORFF, B. F. T.; AGUIAR, D. A.; SILVA, W. F.; SUGAWARA, L. M.; ADAMI, M.; MOREIRA, M. A. Studies on the rapid expansion of sugarcane for ethanol production in São Paulo State (Brazil) using Landsat data. **Remote Sensing**, v. 2, n. 4, p. 1057–1076, 2010. 35

RUSSWURM, M.; KORNER, M. Multi-temporal land cover classification with sequential recurrent encoders. **ISPRS International Journal of Geo-Information**, v. 7, n. 4, p. 129, 2018. 4, 14

RUSSWURM, M.; PELLETIER, C.; ZOLLNER, M.; LEFÈVRE, S.; KÖRNER, M. BreizhCrops: a time series dataset for crop type mapping. **arXiv:1905.11893**, 2020. 14

SANCHES, I. D.; FEITOSA, R. Q.; DIAZ, P. M. A.; SOARES, M. D.; LUIZ, A. J. B.; SCHULTZ, B.; MAURANO, L. E. P. Campo Verde database: seeking to improve agricultural remote sensing of tropical areas. **IEEE Geoscience and Remote Sensing Letters**, v. 15, n. 3, p. 369–373, 2018. 30

SANTOS, L.; FERREIRA, K.; PICOLI, M.; CAMARA, G. Self-organizing maps in earth observation data cubes analysis. In: VELLIDO, A.; GIBERT, K.; ANGULO, C.; MARTIN, J. (Ed.). **Advances in self-organizing maps, learning vector quantization, clustering and data visualization**. [S.l.]: Springer, 2019. p. 70–79. 12

SANTOS, L. A.; FERREIRA, K.; PICOLI, M.; CAMARA, G.; ZURITA-MILLA, R.; AUGUSTIJN, E.-W. Identifying spatiotemporal patterns in land use and cover samples from satellite image time series. **Remote Sensing**, v. 13, n. 5, p. 974, jan. 2021. 12, 13

SCANLON, B. R.; JOLLY, I.; SOPHOCLEOUS, M.; ZHANG, L. Global impacts of conversions from natural to agricultural ecosystems on water resources: quantity versus quality: conversions from natural to agricultural ecosystems. **Water Resources Research**, v. 43, n. 3, mar. 2007. ISSN 00431397. 1

SCHRAMM, M.; PEBESMA, E.; MILENKOVIĆ, M.; FORESTA, L.; DRIES, J.; JACOB, A.; WAGNER, W.; MOHR, M.; NETELER, M.; KADUNC, M.; MIKSA, T.; KEMPENEERS, P.; VERBESSELT, J.; GÖSSWEIN, B.; NAVACCHI, C.; LIPPENS, S.; REICHE, J. The openEO API–harmonising the use of Earth observation cloud services using virtual data cube functionalities. **Remote Sensing**, v. 13, n. 6, p. 1125, jan. 2021. 24, 25

SHEEREN, D.; FAUVEL, M.; JOSIPOVIĆ, V.; LOPES, M.; PLANQUE, C.; WILLM, J.; DEJOUX, J.-F. Tree species classification in temperate forests using Formosat-2 satellite image time series. **Remote Sensing**, v. 8, n. 9, p. 734, sep. 2016. 4

SHIMABUKURO, Y. E.; SANTOS, J. R.; FORMAGGIO, A. R.; DUARTE, V.; RUDORFF, B. F. T. The Brazilian Amazon monitoring program: PRODES and DETER projects. In: HANSEN M. C.; ACHARD, F. (Ed.). **Global forest monitoring from earth observation**. New York: CRC Press, 2012. p. 354. 8, 35

SIMOES, R.; MACIEL, A.; ANDRADE, P.; SANTOS, L.; CAMARA, G.; PICOLI, M. **Source code for: land use and cover change maps for Mato Grosso State in Brazil**. jul. 2019. Zenodo. Available from: <https://doi.org/10.5281/ZENODO.3354400>. Access in: 11 May 2020. 67

SIMOES, R.; PICOLI, M. C. A.; CAMARA, G.; MACIEL, A.; SANTOS, L.; ANDRADE, P. R.; SÁNCHEZ, A.; FERREIRA, K.; CARVALHO, A. Land use and cover maps for Mato Grosso State in Brazil from 2001 to 2017. **Scientific Data**, v. 7, n. 1, p. 34, dec. 2020. ISSN 2052-4463. 2, 14, 17, 27, 48

SOILLE, P.; BURGER, A.; MARCHI, D. D.; KEMPENEERS, P.; RODRIGUEZ, D.; SYRRIS, V.; VASILEV, V. A versatile data-intensive computing platform for information retrieval from big geospatial data. **Future Generation Computer Systems**, v. 81, p. 30–40, apr. 2018. ISSN 0167-739X. 43

SOTERRONI, A. C.; MOSNIER, A.; CARVALHO, A. X. Y.; CAMARA, G.; OBERSTEINER, M.; ANDRADE, P. R.; SOUZA, R. C.; BROCK, R.; PIRKER, J.; KRAXNER, F.; HAVLIK, P.; KAPOS, V.; ERMGASSEN, E.; VALIN, H.; RAMOS, F. M. Future environmental and agricultural impacts of Brazil's Forest Code. **Environmental Research Letters**, v. 13, n. 7, p. 074021, jul. 2018. ISSN 1748-9326. 28

SOUZA, C. M.; SHIMBO, J. Z.; ROSA, M. R.; PARENTE, L. L.; ALENCAR, A. A.; RUDORFF, B. F. T.; HASENACK, H.; MATSUMOTO, M.; FERREIRA, L. G.; SOUZA-FILHO, P. W. M.; OLIVEIRA, S. W. de; ROCHA, W. F.; FONSECA, A. V.; MARQUES, C. B.; DINIZ, C. G.; COSTA, D.; MONTEIRO, D.; ROSA, E. R.; VÉLEZ-MARTIN, E.; WEBER, E. J.; LENTI, F. E. B.; PATERNOST, F. F.; PAREYN, F. G. C.; SIQUEIRA, J. V.; VIERA, J. L.; FERREIRA NETO, L. C.; SARAIVA, M. M.; SALES, M. H.; SALGADO, M. P. G.; VASCONCELOS, R.; GALANO, S.; MESQUITA, V. V.; AZEVEDO, T. Reconstructing three decades of land use and land cover changes in Brazilian biomes with Landsat archive and Earth Engine. **Remote Sensing**, v. 12, n. 17, p. 2735, jan. 2020. 20

SPAROVEK, G.; BARRETO, A. G. O. P.; MATSUMOTO, M.; BERNDES, G. Effects of governance on availability of land for agriculture and conservation in Brazil. **Environmental Science & Technology**, v. 49, p. 10285–10293, 2015. 35

SPERA, S. A.; COHN, A. S.; VANWEY, L. K.; MUSTARD, J. F.; RUDORFF, B. F.; RISSO, J.; ADAMI, M. Recent cropping frequency, expansion, and abandonment in Mato Grosso, Brazil had selective land characteristics. **Environmental Research Letters**, v. 9, n. 6, p. 064010, 2014. 28, 38

STOCKER, B. D.; ROTH, R.; JOOS, F.; SPAHNI, R.; STEINACHER, M.; ZAEHLE, S.; BOUWMAN, L.; XU-RI; PRENTICE, I. C. Multiple greenhouse-gas feedbacks from the land biosphere under future climate change scenarios. **Nature Climate Change**, v. 3, n. 7, p. 666–672, jul. 2013. ISSN 1758-678X, 1758-6798. 1

SUDMANNS, M.; TIEDE, D.; LANG, S.; BARALDI, A. Semantic and syntactic interoperability in online processing of big Earth observation data. **International Journal of Digital Earth**, v. 11, n. 1, p. 95–112, jan. 2018. ISSN 1753-8947. 5

TANDY, J.; BRINK, L. van den; BARNAGHI, P. Spatial data on the web best practices. **W3C Working Group Note**, 2017. 44, 45

TYUKAVINA, A.; HANSEN, M. C.; POTAPOV, P. V.; STEHMAN, S. V.; SMITH-RODRIGUEZ, K.; OKPA, C.; AGUILAR, R. Types and rates of forest disturbance in Brazilian Legal Amazon, 2000–2013. **Science Advances**, v. 3, n. 4, 2017. 28

VERBESSELT, J.; HYNDMAN, R.; NEWNHAM, G.; CULVENOR, D. Detecting trend and seasonal changes in satellite image time series. **Remote Sensing of Environment**, v. 114, n. 1, p. 106–115, 2010. 4

VINHAS, L.; RIBEIRO, G.; FERREIRA, K. R.; CAMARA, G. Web services for big Earth observation data. In: BRAZILIAN SYMPOSIUM ON GEOINFORMATICS, 17., 2016, Campos do Jordao. **Proceedings...** Campos do Jordão, SP, Brazil, 2016. p. 26–35. 31

WALTER, B. M. T. **Fitofisionomias do bioma Cerrado: síntese terminológica e relações florísticas.** PhD Thesis (PhD in Ecology) — Universidade de Brasilia, Brasília, 2006. 19

WICKHAM, H. **Advanced R**. 2. ed. Boca Raton: Chapman and Hall/CRC, 2019. ISBN 978-0-8153-8457-1. 18

_____. **The tidy tools manifesto**. 2019. Available from: <https://cran.r-project.org/web/packages/tidyverse/vignettes/manifesto.html>. Access in: 26 Sep 2020. 45

WIENS, T. S.; DALE, B. C.; BOYCE, M. S.; KERSHAW, G. P. Three way k-fold cross-validation of resource selection functions. **Ecological Modelling**, v. 212, n. 3, p. 244–255, 2008. ISSN 0304-3800. 38

WOODCOCK, C. E.; LOVELAND, T. R.; HEROLD, M.; BAUER, M. E. Transitioning from change detection to monitoring with remote sensing: a paradigm shift. **Remote Sensing of Environment**, v. 238, p. 111558, mar. 2020. ISSN 00344257. 5

WULDER, M. A.; HERMOSILLA, T.; STINSON, G.; GOUGEON, F. A.; WHITE, J. C.; HILL, D. A.; SMILEY, B. P. Satellite-based time series land cover and change information to map forest area consistent with national and international reporting requirements. **Forestry: An International Journal of Forest Research**, v. 93, n. 3, p. 331–343, may 2020. ISSN 0015-752X. 1

WULDER, M. A.; MASEK, J. G.; COHEN, W. B.; LOVELAND, T. R.; WOODCOCK, C. E. Opening the archive: how free data has enabled the science and monitoring promise of Landsat. **Remote Sensing of Environment**, v. 122, p. 2–10, 2012. 1, 3

ZHONG, L.; LINA-HU; YU, L.; GONG, P.; BIGING, G. S. Automated mapping of soybean and corn using phenology. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 119, p. 151–164, 2016. ISSN 0924-2716. 39

ZHU, Z.; ZHANG, J.; YANG, Z.; ALJADDANI, A. H.; COHEN, W. B.; QIU, S.; ZHOU, C. Continuous monitoring of land disturbance based on Landsat time series. **Remote Sensing of Environment**, v. 238, p. 111116, 2020. 4

APÊNDICE A - CODE AVAILABILITY

The code scripts that support the findings of Chapter 2 are openly available in *sitsdata* repository at GitHub (https://github.com/e-sensing/sitsdata).

The code scripts that support the findings of Chapter 3 was published by (SIMOES et al., 2019) and is available at https://doi.org/10.5281/zenodo.3354400 under the GNU General Public Licence v3.0. Unfortunately, the code refer to an earlier version of *sits* and the MOD13Q1 collection 6 images repository referred therein are no more open for the public access. Adaptations in the code script may be required.

The *sits* package is available at Github (https://github.com/e-sensing/sits) under the GNU General Public License v3.0. Full documentation of the package is available at https://e-sensing.github.io/sitsbook/ Web book.

The *rstac* package is available at the Comprehensive R Archive Network (CRAN) in the address https://cran.r-project.org/package=rstac. The software is licensed under the MIT License. The development version of the package is available at BDC project repository in GitHub (https://github.com/brazil-data-cube/rstac)