GHENT
UNIVERSITY

Faculty of Sciences

Modeling of climate-vegetation dynamics using machine learning techniques in a non-linear Granger causality framework

Thomas Mortier

Master dissertation submitted to
obtain the degree of
Master of Statistical Data Analysis

Promotor: prof. dr. Willem Waegeman
Co-promotor: prof. dr. ir. Olivier Thas
Tutor: Stijn Decubber

Department of Applied Mathematics, Computer Science and Statistics

Academic year 2016 - 2017

The author and the promoter give permission to consult this master dissertation and to copy it or parts of it for personal use. Each other use falls under the restrictions of the copyright, in particular concerning the obligation to mention explicitly the source when using results of this master dissertation.

Thomas Mortier
June, 2017

Foreword

This thesis was carried out as part of the SAT-EX project, under supervision of the KERMIT research group, and as a continuation of research by Stijn Decubber and Christinna Papagiannopoulou.

First, I would like to thank my promotor prof. dr. Willem Waegeman and co-promotor prof. dr. ir. Olivier Thas, for giving me the opportunity to work on this thesis, allowing me to combine statistical methods with deep learning techniques.

Furthermore, I would like to thank Stijn Decubber for the numerous discussions and support throughout this thesis. Again, I would like to thank Stijn Decubber and prof. dr. Willem Waegeman, for the reading and supervision of this work.

More importantly, I would like to express my deepest gratitude to my family and friends. Moreover, I would like to thank my parents, for giving me the opportunity to further expand my knowledge and understanding of statistical data analysis. A special thanks also goes to Charlotte, for the unconditional love, motivation and consolation throughout these years. Without them, this work would not have been possible.

Upon writing this last sentence, I am not only finishing my second and final thesis, but also my life as a student, at the University of Ghent.

Thomas Mortier
June, 2017

Table of Contents

1 Introduction **1**

1.1 Climate change research . . . 1

1.1.1 Forecasting . . . 2

1.1.2 Attribution . . . 2

1.2 Climate change modeling . . . 2

1.2.1 Mechanistic models . . . 3

1.2.2 Data-driven models . . . 3

1.2.3 Causal inference . . . 3

1.2.3.1 Causation versus association . . . 4

1.2.3.2 Granger causality . . . 5

1.3 Summary and goals . . . 7

2 Data and methods **9**

2.1 Data description and exploration . . . 10

2.1.1 Raw data . . . 11

2.1.1.1 Normalized Difference Vegetation Index (NDVI) . . . 11

2.1.1.2 Leaf Area Index (LAI) . . . 11

2.1.1.3 Cyclic and seasonal behaviour . . . 12

2.1.2 Residual data . . . 13

2.2 Machine learning techniques . . . 18

2.2.1 (Un)supervised learning . . . 18

2.2.2 Bias-variance tradeoff . . . 18

2.2.3 Artificial neural networks . . . 20

2.2.3.1 Feedforward neural networks . . . 20

2.2.3.2 Convolutional neural networks . . . 21

2.2.3.3 Data-driven modeling with CNNs . . . 23

2.3 Non-linear Granger causality framework . . . 24

3 Results **27**

3.1 Data preprocessing . . . 27

3.2 Model building . . . 30
3.2.1 Convolutional/locally recurrent networks . . . 30
3.2.2 Convolutional LSTM networks . . . 32
3.2.3 Temporal convolutional networks . . . 33
3.3 Model selection and discussion . . . 34
3.3.1 Architecture and (hyper)parameters . . . 34
3.3.2 Evaluation . . . 36
3.3.2.1 Training, validation and test results . . . 36
3.3.2.2 MSE analysis . . . 39
3.3.2.3 BMWU analysis . . . 42

4 Conclusion **45**
4.1 Summary . . . 46
4.2 Future work . . . 47

References **48**

A Software and hardware specifications **55**

B Additional results **57**
B.1 MSE analysis . . . 57
B.2 BMWU analysis . . . 62
B.2.1 General . . . 62
B.2.2 Monthly . . . 63

Abstract

Due to recent improvements in satellite technology and increasing number of global historical records of environmental and climatic variables, research in climatology has become more valuable and crucially important. Furthermore, the huge amount of data provides new ways to start unravelling underlying processes, which drive long-term changes in climate extremes or allow researchers in trying to understand the impact of latter changes on terrestrial ecosystems. However, as the amount of available data in climate science, or more specific climate change attribution research, is increasing exponentially, the need for better/complex data-driven models is crucial in order to allow new opportunities for research and industry, as well as gathering novel insights.

As part of the SAT-EX project, under supervision of the KERMIT research group, this thesis is a continuation on comprehensive research by Decubber S., Papagiannopoulou C., et al. [6] [23]. We will mainly focus on climate change attribution research, where the influence of various climatological extremes such as temperature and precipitation on vegetation will be studied. Moreover, we will extend current state of the art research by using more complex machine learning models such as deep neural networks, within a non-linear Granger causality framework [23], in order to exploit and analyse various G-causal relationships between climatological variables and vegetation. Furthermore, in this thesis the modeling of climate data will be improved by exploiting neighbouring information, together with an improved global scale learning approach. For the latter, deep learning techniques can be ideally used as an alternative to less flexible statistical models. In order to deal with the black box problem, that comes with various deep learning models, a compromising statistical-deep learning approach will be considered, by using deep learning techniques within a non-linear Granger causality framework.

Keywords: climate science, climate change attribution, forecasting, (non-linear) Granger causality test, statistical learning, deep learning, convolutional neural networks, recurrent neural networks, long short-term memory networks

1 Introduction

Due to recent improvements in satellite technology and increasing number of global historical records of environmental and climatic variables, research in climatology has become more valuable and crucially important. As of today, the huge amount of climatological data provides new ways to start unravelling underlying processes which drive long-term changes in climate extremes or allow researchers in trying to understand the impact of latter changes on terrestrial ecosystems. In contrast to understanding various climatic relationships, climatic data also allows for benchmarking in order to evaluate the skills of various climatic models.

In this brief introduction we will guide the reader through fundamental aspects in climate sciences and give an overview on the general structure of this thesis. We start with a formal definition of climate change research, by looking at two different subdomains which will be explained in the following section. We then continue this chapter by looking at climate change modeling techniques, whereafter we end this introduction with a brief summary and overview of this thesis.

1.1 Climate change research

Generally speaking, we can divide climate change research in the following topics:

- Climate change forecasting
- Climate change attribution

We will not put a lot of focus on the discussion of these two domains, but want to emphasize the main difference between the two and focus mostly on climate change attribution, the central focus in this research.

1.1.1 Forecasting

In the domain of climate change forecasting, one will typically study forecasting methods where we are interested in predicting some future state of the climatic system. However, important to mention is the distinction between weather and climate. When it comes to weather progression, we are intuitively familiar with the daily changes in temperature, rain which comes and goes, or some severe storm that is predicted to hit in one of the upcoming days. Characteristic time scale for changes in weather generally depend on latitude, e.g. in the tropics, the weather tends to be much steadier, with sunny periods and steady trade winds punctuated by a short daily downpour. At the other hand, the concept of climate is also familiar, as we typically recall that warm summer some years ago or that snowy winter. Hence, the reader should already notice the subtle difference between climate and weather: climate is the statistics of weather averaged over a large-scale time period that contains many weather events.

1.1.2 Attribution

Changes in climatic means are either due to a small change acting over the entire averaging period or by the unusual occurence of extreme events within the particular averaging period. Since climate refers to the statistics of the atmosphere, which interacts strongly with the surface through interchange of heat, momentum and water, the climatic state of the atmosphere therefore depends strongly on the state of the surface. The latter can be characterized by its temperature, reflectivity, surface moisture etc. Due to this interaction, surface conditions will generally change, causing atmospheric statistics to change in response and the other way around. Hence, in climate change attribution, researchers are more interested in identifying and quantifying cause-effect relationships between atmospheric statistics (e.g. precipation or temperature) and other climate variables (e.g. surface characteristics such as vegetation), rather than predicting long-term future states. In this thesis we will model climate-vegetation dynamics in order to analyse cause-effect relationships between climatic features and vegetation, which corresponds to climate change attribution.

1.2 Climate change modeling

As we've already discussed two different research topics in climate sciences, one also needs appropriate tools in order to allow forecasting or identifying and quantifying cause-effect relationships within climatic systems. Moreover, in the context of this research (i.e. climate change

attribution), different modeling techniques are available which will be discussed in the subsequent sections. After modeling climate-vegetation dynamics, one would then need appropriate statistical inference techniques in order to identify and reason about cause-effect relationships (see Section 1.2.3). We start by giving the reader a brief overview of two general modeling approaches [9].

1.2.1 Mechanistic models

Although many approaches currently exist, one of the most standard approaches currently used are based on simulation studies with mechanistic climate models. These models represent hypothesized relationships between various variables, where the relationship is specified by means of biological processes, typically formalized through differential equations. The parameters all have physical meanings and can be measured independently of the dataset. They are designed to reflect our hypothesized understanding of physical reality.

1.2.2 Data-driven models

In contrast to the latter concept-based models, data-driven models assume no underlying physical representation of reality. Here we will model the relationships by learning flexible functions of some set of input data. Hence, these (statistical) models are directly used, without any prior knowledge, on a given set of data.
As already stated in the beginning of this chapter, recent improvements in satellite technology as well as in-situ technology, caused a tremendous increase in global observations on finer spatio-temporal resolutions. One can see that the resulting amount of big data can be ideally used by data-driven models, in order to answer questions in climate change attribution research. the only question remaining is whether data-driven models can handle the complex nature of the latter data, and whether more complex *deep learning* models can be used as an alternative to less complex statistical models. As for now, we emphasize the choice for data-driven models in this thesis.

1.2.3 Causal inference

Within climate change attribution, after modeling some climatic dynamics, one needs to make conclusions on whether cause-effect relationships exist or not. Hence, we need a more formal definition of cause-effect relationships and their characterstics. Cause and effect (also referred to as causation or causality) is an abstraction that indicates how the world progresses, how one process (or *cause*) is connected to another process or state (or *effect*). As an example, in this thesis we will focus on how changes in climatic processes (e.g. temperature, precipation) cause an effect on another process (e.g. vegetation). Causality is a term that is widely used in many domains, such as science, metaphysics, management, humanities, theology, etc. Debating about

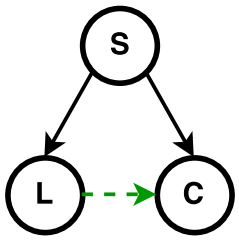


Figure 1.1: Making the erroneous statement that carrying a lighter (L) causes cancer (C), is obscured by observing the true cause smoking (S). We observe that carrying a lighter is only associated (green dashed line) with developing cancer.

causality would require another few chapters, if not books, hence we will only focus on causal inference that is widely used in many scientific domains. Before going into different causal inference techniques, we first need to emphasize a crucial difference between two concepts that are commonly used, namely *causation* and *association*.

1.2.3.1 Causation versus association

The reader should be familiar with the fact that **association does not imply causation**. For instance, a researcher found a significant relationship between carying a lighter and developing lung cancer during the middle age. That is, the likelihood of developing lung cancer significantly increased when carrying a lighter, in contrast to people who did not carry a lighter (on daily basis). The researcher could argue that carrying a lighter causes an effect, in terms of developing lung cancer or that a causal relationship exists between the two. Putting it simply, this is exactly what we mean with "association does not imply causation": the researcher may observe a statistically significant association (green dashed line in Figure 1.1) between carrying a lighter (L) and developing cancer (C), but needs to be aware of the fact this does not imply causation. Indeed, after observing a new variable/state smoking (S), we may intuitively conclude that the latter relationship is due to a common state (S). Formally speaking, the observation of an effect/change in state (i.e. developing lung cancer) due to carrying a lighter is obscured by a common cause (i.e. smoking). When it comes to inference, one also needs to distinguish between *causal inference* and *inference of association*. The former analyses the response of the effect variable when the cause is changed.
Hence, causal inference is very complex and needs appropriate mathematical formalization in combination with important assumptions, in order to use in a statistical modeling framework. With the latter assumptions we may refer to model-related assumptions (e.g. assumptions in normal error linear regression models) and the even more important assumptions that come with causality:

- Identify and include all possible common causes (e.g. Figure 1.1).
- Identify and include all possible confounders.

Important to mention is that the inclusion of common causes or confounders could obscur or change direct observed causal relations.
Hence, we want to empasize that causal inference and corresponding assumptions, in climate change attribution research, are mostly not fulfilled due to the complex nature of the data. Taking in account all possible common causes or confounders is a tremendously difficult task, for which additional domain expertise is necessary. When the latter expertise is not available, one can always assume the causal relationship as a hypothesis that can be further investigated, while realising that the relationship is potential and relative to the amount of information that is available in the data [7].

1.2.3.2 Granger causality

As already mentioned, causal inference needs appropriate mathematical formalization for which various techniques exist. Probabilistic graphical models (PGMs), where one is interested in discovering direct probabilistic edges between variables in some graph, are widely used models where causal inference can be drawn and visualized in a graphical way. Although these models are very popular and allow for graphical representations and reasoning about a given dataset, they become very complex when dealing with high resolution spatio-temporal datasets (e.g. climate change attribution data). As of today, the latter issues have already been addressed by using extended formulations of PGMs [20][21].

As an alternative to PGMs, we will focus on causal discovery by using the concept of *Granger causality*. This technique has risen from the field of econometrics and was invented by Nobel Prize winner Clive Granger [12]. Due to its computational simplicity, it remains a popular method for causal inference in temporal data. Intuitively we can say that a variable X (which evolves over time) *Granger-causes* another evolving variable Y if predictions of Y, with the inclusion of its own past values and on the past values of X, are better than predictions of Y based only on its own past values. We have to emphasize the fact that Granger causality (also referred to G-causality), may not be mistaken with the "true causality" concept, as we've already seen. Hence, G-causality does not always imply true causality and is based on two principles:

- The cause happens prior to its effect.
- The cause has unique information about the future values of its effect.

In multivariate analysis (or more specific in multivariate economic time series), Granger causality is usually performed by fitting a *vector autoregressive model* (VAR) to some multivariate

time series. For instance, let $X(t) \in \mathbb{R}^{d\times 1}$, for $t = 1, ..., T$, be a d-dimensional multivariate time series. That is, $X(t)$ represents d different variables evaluated in time. Granger causality is then performed by fitting a VAR model with L time lags as follows

$$X(t) = \sum_{\tau=1}^{L} A_\tau X(t-\tau) + \epsilon(t), \tag{1.1}$$

where we assume multivariate normal errors $\epsilon(t)$, and for each τ we have

$$A_\tau = \begin{bmatrix} \beta_{11,\tau} & \dots & \beta_{1d,\tau} \\ \vdots & \ddots & \vdots \\ \beta_{d1,\tau} & \dots & \beta_{dd,\tau} \end{bmatrix}. \tag{1.2}$$

The Granger causality test is then applied by looking if at least one of the coefficients $A_\tau(j, i)$, for $\tau = 1, ..., L$, is statistically significant larger than zero (in absolute value) [15]. In terms of predictive modeling, one could also test a G-causal relation between some feature of interest $X(t)$ and a response variable $Y(t)$, by means of comparing the predictive performance of two nested models

$$\begin{aligned} \hat{Y}_b(t) &= f_b(y_{t-1}, ..., y_{t-L}), \\ \hat{Y}_e(t) &= f_e(y_{t-1}, ..., y_{t-L}, x_{t-1}, ..., x_{t-L}), \end{aligned} \tag{1.3}$$

where we denote the baseline model by f_b and f_e by extended model respectively. Using the defined models, the null hypothesis of Granger non-causality would be formulated as the null hypothesis that f_b and f_e would yield the same prediction error, whereas the alternative is one-sided, such that if f_e predicts significantly better than f_b.

1.3 Summary and goals

As part of the SAT-EX project[1], under supervision of the KERMIT research group, this thesis is a continuation of comprehensive research by Decubber S., Papagiannopoulou C., et al. [6] [23]. We will mainly focus on climate change attribution research, where the influence of various climatological extremes such as temperature and precipitation on vegetation will be studied. Moreover, we will extend current state of the art research by using more complex machine learning models such as deep neural networks, within a non-linear Granger causality framework [23], in order to exploit and analyse various G-causal relationships between climatological variables and vegetation. Furthermore, the modeling of climate data can be improved by exploiting neighbouring information, hence one might consider multitask learning approaches that learn from multiple locations simultaneously [6][23]. However, due to the high dimensional ($p >> n$) nature of climate data, the complexity of the latter models increases exponentially. In order to solve the latter problem, we will propose and study various deep neural networks within the non-linear Granger causality framework. Nowadays, neural networks have already been proven to be succesful in many pattern recognition domains such as speech recognition, image recognition etc. Although deep neural networks are highly complex and powerful models, they come at the expense of being blackbox in nature. In this thesis, we will try to combine the powerful but blackbox nature of neural networks, together with the non-linear Granger causality framework, in order to discover various patterns in climatic datasets. However, we want to emphasize that no focus will be put on drawing conclusions on causal relationships between different climate variables.

The following chapters are summarized as follows:

- Chapter 2 (data and methods): we will briefly discuss the datasets that we have been using throughout this work, providing the reader with some exploratory analysis results. Finally, the general methodology will be discussed.
- Chapter 3 (results): this chapter will further focus on the applied techniques and obtained results, by using the methodology, as discussed in Chapter 2.
- Chapter 4 (conclusions): in this chapter, we conclude with a general summary and conclusion.

[1] SAT-EX Project homepage: http://www.sat-ex.ugent.be/

2

Data and methods

Now that the reader is familiar with the basic concepts in climate change research, or moreover climate change attribution, together with modeling and causal inference techniques, we will now continue with the discussion of the data that was available for this thesis together with the discussion of the used methods. As already stated in the previous chapter, drawing conclusions on causal relationships between different climate variables will not be handled in this thesis. However, in the context of this thesis we will study a potential candidate framework in which we move towards a more deep learning data-driven modeling approach (see Section 1.2.2), together with a non-linear Granger causality framework for detecting potential causal relations. Again we want to emphasize that Granger causality does not imply true causality.
Although different approaches currently exist, our general framework can be visually illustrated as seen in Figure 2.1. Climate data will be used as input for data-driven models (i.e. deep learning models in this thesis), whereafter various outputs from these models (e.g. predicted

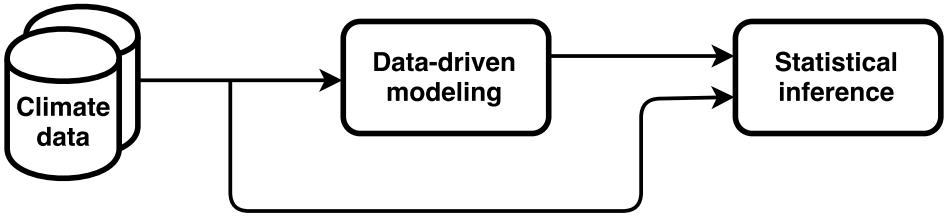


Figure 2.1: General workflow in data-driven climate change attribution. Climate data is used as input for various data-driven models. Climate data is then used, together with various outcomes from the modeling step, in a statistical inference framework in order to obtain insights.

variables, predictions errors, etc.) will be used, together with the raw climate input data, in a last statistical inference step (e.g. non-linear Granger causal inference). One could argue that data-driven modeling can be categorised under statistical inference (i.e. leaving the above workflow with only one processing step), but in the context of this thesis we want to distinguish between the deep learning modeling approach and the statistical inference step. By using powerful deep learning models, combined with statistical inference techniques, we would like to gain more and better insights into climate dynamics. The central question, whether deep learning models can be used as data-driven models in this framework, will be discussed in this thesis.
We start with a summary of the available data, together with some brief exploratory analysis (see Section 2.1). Finally, we discuss the general methods and techniques, that have been applied within the context of this thesis (see Section 2.2 and 2.3).

2.1 Data description and exploration

As already stated in the introductory chapter (Section 1.3), this thesis is a continuation of previous research, in which the used datasets were composed and provided by C. Papagiannopoulou, S. Decubber, et al. (KERMIT, department of Mathematical Modelling, Statistics and Bioinformatics, Faculty of Bioscience Engineering) [6][23]. By using several publicly available satellite datasets, covering different spatial areas, different temporal intervals and various resolutions, multiple datasets were assembled. Although different datasets were obtained for various features such as temperature, precipitation, soil moisture, snow depth, radiation, vegetation etc., we will only provide the reader with a general overview of the data that was eventually used in this thesis.

Variable (unit)	**Source (code)**	**Spatial res.**	**Temporal res.**	**Coverage**
Temperature (K)	ERA	0.25°	daily	1979-2013
Precipitation (mm)	MSWEP	0.25°	daily	1982-2012
Radiation short (W/m^2)	ERA	0.25°	daily	1979-2013
Greenness index (NDVI)	GIMMS	0.25°	monthly	1982-2012
Leaf area index (LAI)	**GLASS**	**0.25°**	**9 days**	**1982-2012**

Table 2.1: Overview of the used datasets in this thesis. Note that we will mainly focus on the LAI dataset (i.e. a measurement of vegetation), as outcome variable of interest, due to the finer temporal resolution compared to other data products related to vegetation such as NDVI. The source code refers to the source for each dataset.

2.1.1 Raw data

The complete dataset can be seen as a collection of different dataframes on different features (e.g. temperature, precipitation, etc.), for which most dataframes (see Table 2.1) contain data on 180×360 pixels. The latter dataframes are originally stored in HDF5-format[1] files, allowing for efficient and fast I/O processing. Depending on the feature, each dataframe further contains temporal information on daily, monthly, etc. basis. Hence, each dataframe can be seen as a collection of 180×360 time series, or more formal as a multivariate time series. Since we are interested in studying the influence of various features on vegetation (see Section 1.3), by using the (non-linear) G-causal inference idea as described in (1.3), we hence need an appropriate vegetation measurement that can be used as outcome variable of interest. In Table 2.1, we've listed two possible dataframes, which both represent data on vegetation.

2.1.1.1 Normalized Difference Vegetation Index (NDVI)

The first dataframe contains Normalized Difference Vegetation Index (NDVI) data, which can be used as a proxy for the amount of vegetation. The latter represents a graphical indicator that uses the spectral reflectance measurements in the visible (VIS) and near-infrared (NIR) regions. Without going to deep into technical details, the rationale behind NDVI is based on the fact that vegetation absorbs visible light and at the same time reflects light in the near-infrared region, in order to protect itself against overheating. Hence, the domain of NDVI is restricted to $[-1, 1]$ and higher values of NDVI indicate a higher density of (green) vegetation. For instance, when it comes to tropical rain forests, maximal NDVI values are expected.

2.1.1.2 Leaf Area Index (LAI)

In contrast to NDVI, one could also use Leaf Area Index (LAI) data as a proxy for vegetation, which is highly correlated with NDVI. This dimensionless quantity characterizes plant canopies (i.e. aboveground portion of a plant community or crop) and is defined as the one-sided green leaf area per unit ground surface area (leaf area (m^2)/ground area (m^2)) in broadleaf canopies [26]. We will not further discuss the interpretation and rationale behind this metric, but however do note that the range for LAI is restricted to $[0, < 20]$. Again, higher values indicate dense and green vegetation. In order to keep the temporal resolution for each feature as small as possible, we've chosen not to work with NDVI, as the latter was only available on a monthly temporal resolution. The remaining features, in Table 2.1, will not be further discussed as the meaning/interpretation is straightforward.

[1] HDF5 homepage: https://www.hdfgroup.org/

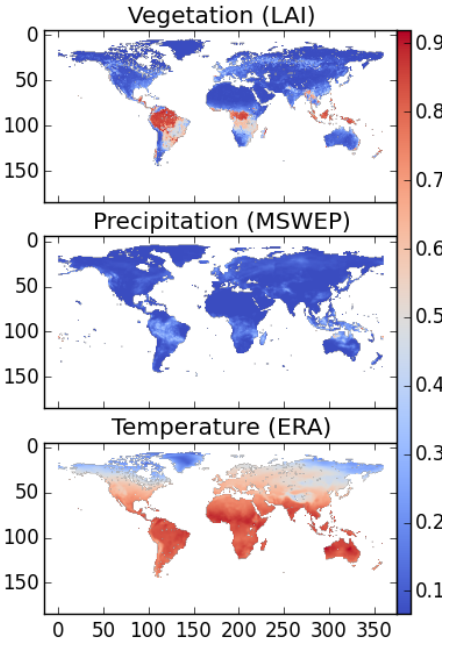
2.1.1.3 Cyclic and seasonal behaviour

In Figure 2.2, three features (i.e. vegetation, precipitation and temperature) are visualised in the corresponding spatial and temporal domain. It is clear that high vegetation values are less spread out (e.g. significant high vegetation is observed in the southern hemisphere), in contrast to temperature. Hence, it seems that higher values for vegetation, as well as for precipitation, are more densely clustered, whereas temperature values are more spread out across the spatial dimensions.

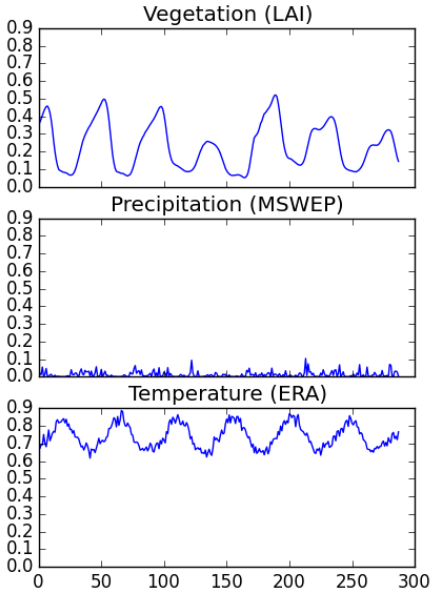
While analysing the features in the temporal domain, the reader should already notice the non-stationary behaviour for vegetation and temperature. More formally, let $\{X_t\}$ be a stochastic process with $F_X(x_{t_1+\tau}, \ldots, x_{t_k+\tau})$, the cumulative distribution function of $\{X_t\}$, evaluated at time points $t_1, \ldots, t_k + \tau$. It can be shown that $\{X_t\}$ is said to be strictly stationary if

$$F_X(x_{t_1+\tau}, \ldots, x_{t_k+\tau}) = F_X(x_{t_1}, \ldots, x_{t_k}), \quad \forall k, \tau, t_1, \ldots, t_k. \tag{2.1}$$

Moreover, as τ does not affect $F_X(.)$, we can say that F_X does not depend on time. For instance, in Spain, we expect high temperatures during the summer and the opposite during the winter, which can be seen in Figure 2.2b. Besides the cyclic pattern, we would also observe an additional increasing trend in a larger time domain, which can be explained by the influence of climate change and global warming. As we are interested in a framework, allowing for infer-



(a) Feature maps evaluated at a fixed time point.



(b) Feature values at fixed location (i.e. Spain; 39.41,-4.77), in function of time.

Figure 2.2: Visualisation of feature maps, evaluated at some arbitrary point in time, and fixed feature map locations evolving over time. The datasets were normalised to $[0, 1]$ *range.*

ence and analysing the influence of different features on vegetation, one needs to be aware of the latter non-stationary processes with cyclical components, underlying the different features that we have discussed. In the particular context of (non-linear) G-causal inference, by means of prediction models as defined in (1.3), one may already expect that a simple (e.g. linear regression) baseline model would already easily capture the underlying cyclic trend in vegetation, resulting in accurate predictions. Besides that, recalling the Granger causality assumptions, we know that stationarity is an assumption which needs to be fulfilled. We might argue that important information, which resides as noise in the cyclic trended signals, would be difficult to capture. At the other hand, by increasing the complexity of the latter models, capturing the latter informative "noise" could be feasible.
As for now, the reader should understand the behaviour and underlying nature of the various features, as well as the issue that we have discussed, concerning the non-stationary property. Nevertheless, we will come back at this issue in the following sections.

2.1.2 Residual data

As we recall the previous section, a lot of features (e.g. temperature, vegetation) seem to follow a seasonal cycle with an additional underlying trend. As this property is intuitively explained by the underlying nature and climate dynamics, it might be useful to analyse whether additional information can be obtained, next to the cyclic and trend components of the multivariate time series. Moreover, in earlier research by Decubber S., Papagiannopoulou C., et al. [6] [23], raw time series were decomposed in terms of anomalies, using an additive linear approach. Each time series $X^T(t)$, is de-trended over an entire study period by modeling the trend $\tilde{X}^T(t)$, using a linear model, with time t as predictor variable. Formally, this can be described as

$$\tilde{X}^T(t) = \alpha_0 + \alpha_1 t. \tag{2.2}$$

The de-trended time series are then obtained by subtracting the trend (2.2) from the original time series $X^T(t)$. Next, the seasonal cycle $\tilde{X}^C(t)$ is estimated by computing the monthly averages over the entire study period, whereafter the de-trended time series $X^T(t) - \tilde{X}^T(t)$ are finally subtracted with the seasonal cycle component $\tilde{X}^C(t)$, yielding the final anomalies

$$R(t) = X^T(t) - \tilde{X}^T(t) - \tilde{X}^C(t). \tag{2.3}$$

This technique is illustrated in Figure 2.3 and 2.4.
As trends and cycles are removed, it is not difficult to see that a baseline model, as in (1.3), would yield lower prediction performances. As a matter of fact, we might expect that more information would be necessary, next to past states of residual vegetation, in order to predict future states of vegetation. For instance, analysing the (cor)relation between NDVI residuals and temperature residuals (for the antecedent month), is depicted in Figure 2.5. Looking at

regio Spain, during the summer, we observe a significant negative correlation with temperature residuals from the antecedent month. This is not surprising, as a particular warm spring in Spain will most likely negatively influence the outcome of vegetation state, during the summer. Nevertheless, beside the observed significant correlations, we also observe that the relationship between temperature and vegetation generally depends on the season (i.e. temporal), as well as on location (i.e. spatial). Consequently, it seems that residuals contain valuable information, which can be exploited. Moreover, one might expect that the extended model (i.e. temperature included), would outperform the baseline model, as additional valuable information is contained within the temperature residuals. Finally, in terms of G-causal inference, as described in (1.3), we can argue on whether raw data or residual data is more appropriate to use. It is clear that a relatively simple baseline model would already yield high predictive performance on the raw vegetation data and hence, one should question whether comparing the corresponding baseline model with a more complex extended model would yield statistically relevant improvements. Besides that, it is not unlikely that to some extent, each model would only discover and learn the underlying trend and cycles, which is of course unwanted in the context of climate change attribution research. At the other hand, valuable information might be lost, while obtaining the residual data, and thus we may prefer to work on all the information that is available, together with more complex models. In this thesis, we will focus on the use of deep learning models (which can be used on spatial global scale) and raw data only, as opposed to previous research [6] [23]. By controlling on overfitting, we will analyse whether these complex models are able to learn the cycles and trends, underyling the data, together with the more important residual information, which silently resides within the raw data. As for now, we only want to emphasize the difference between raw and residual data, together with their corresponding consequences on climate change attribution research.

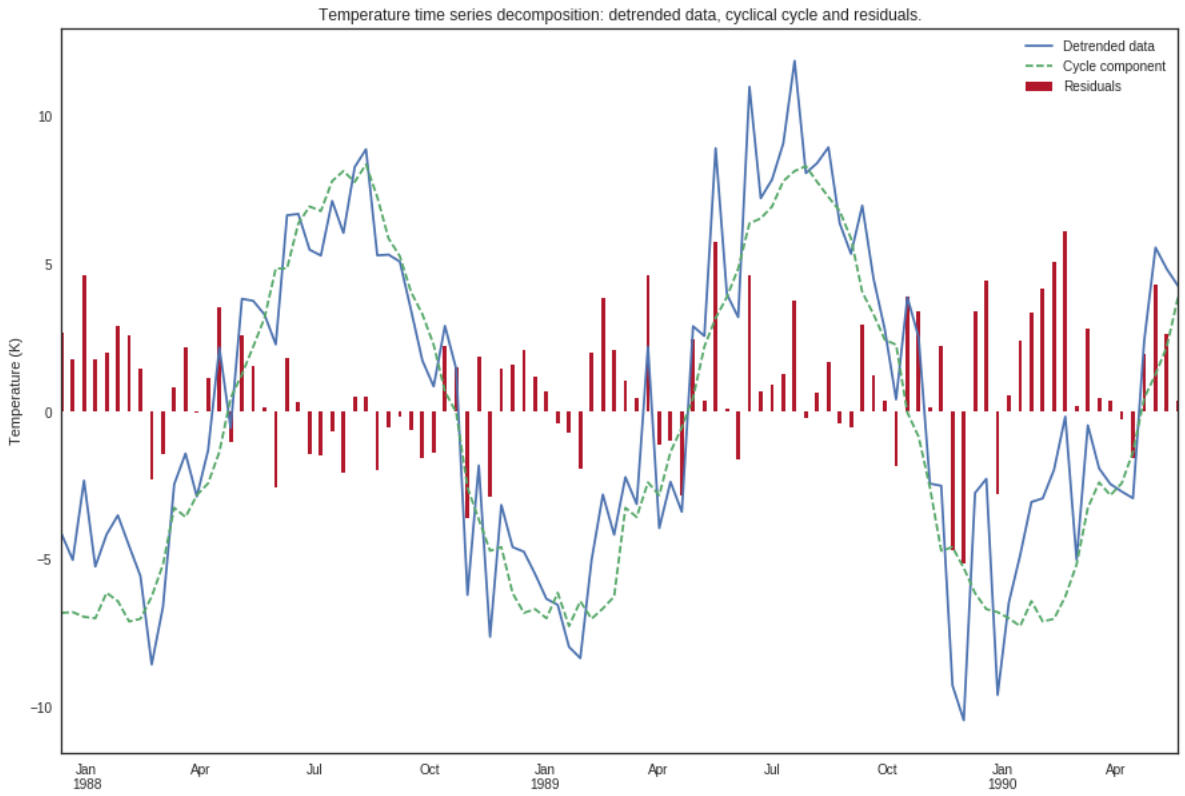


Figure 2.3: Visual representation of temperature time series decomposition. The blue line corresponds to the de-trended temperature time series $(X^T(t) - \tilde{X}^T(t))$*, the green dashed line indicates the seasonal cycle component* $\tilde{X}^C(t)$ *and the red bars illustrate the final obtained residuals* $R(t)$*. Image obtained by Decubber S.*

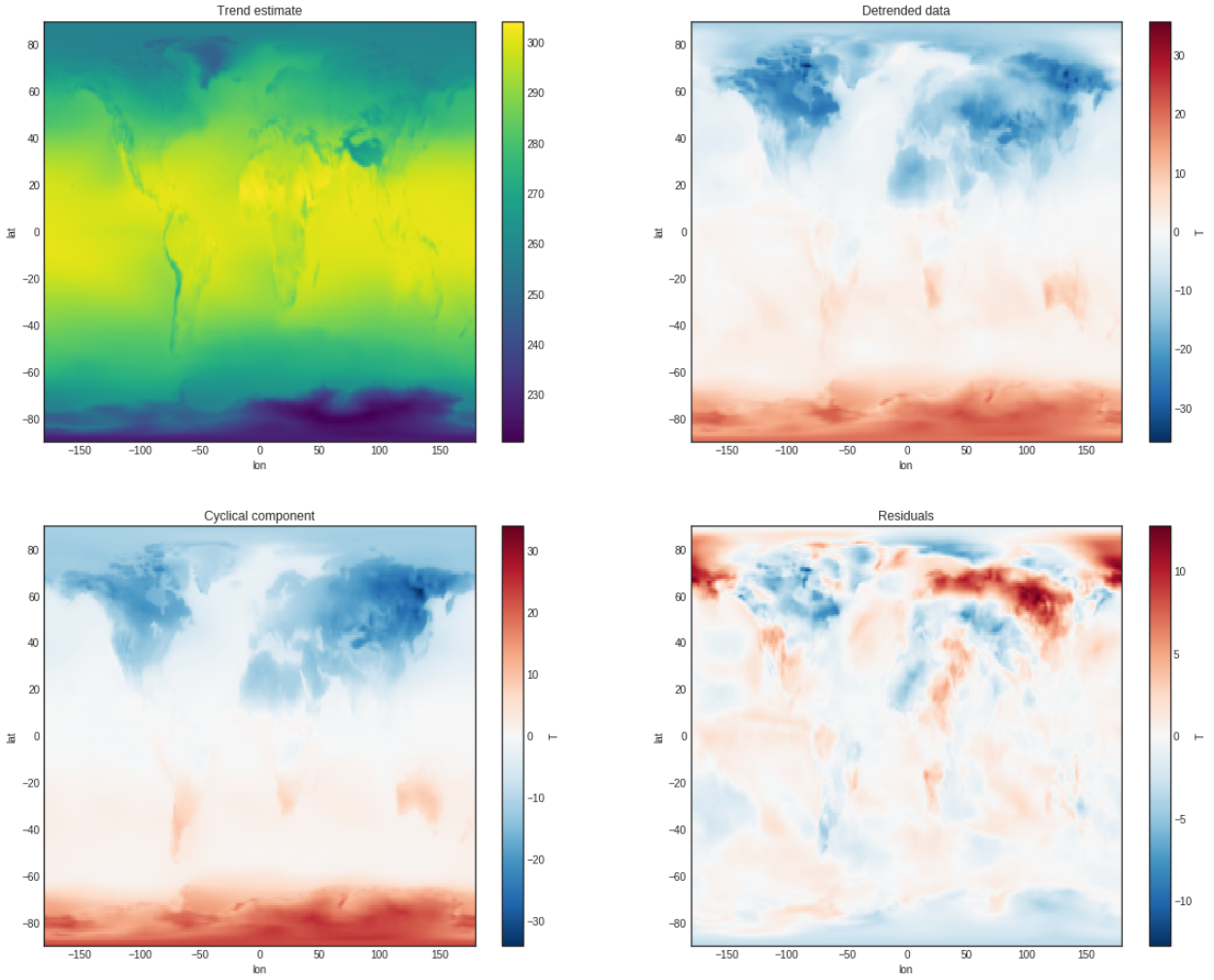


Figure 2.4: Processed temperature maps, corresponding to four different steps in the temperature time series decomposition, evaluated for an arbitrary chosen timepoint t. In the upper left corner the trend estimate $\tilde{X}^T(t)$ is illustrated together with the de-trended temperature map $(X^T(t) - \tilde{X}^T(t))$, as illustrated in the upper right corner. From left to right, the bottom maps correspond to the seasonal cycle component $\tilde{X}^C(t)$ and residuals $R(t)$, respectively. Image obtained by Decubber S.

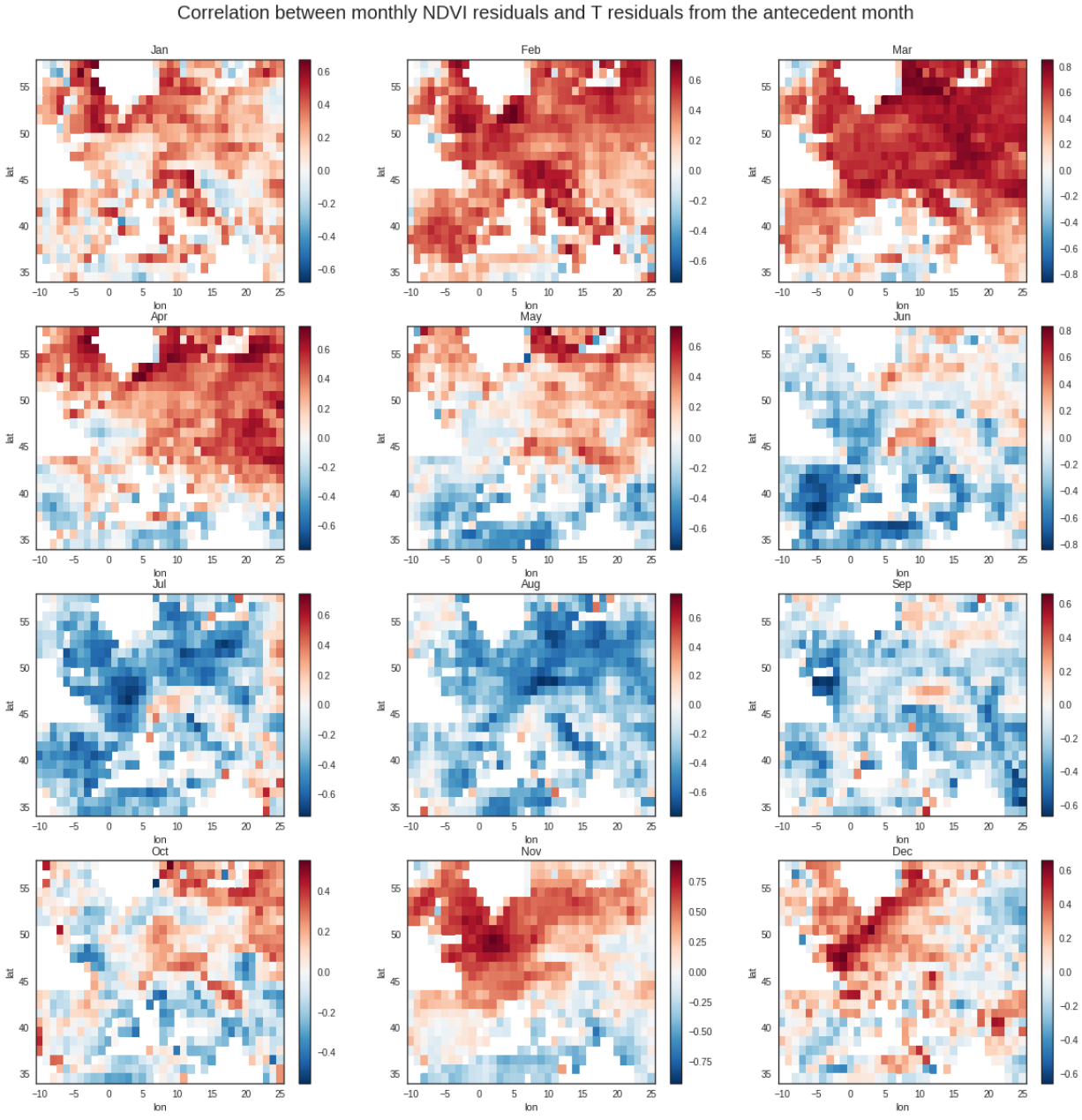


Figure 2.5: Correlation between NDVI (vegetation) and temperature residuals from the antecedent month, evaluated for Europe and each month. Positive correlation (denoted as red), observed in the upper three images, indicates that high temperature during the winter, result in more vegetation for January, February and March. Moreover, the opposite seems to be true, during summer. Image by obtained Decubber S.

2.2 Machine learning techniques

In this section, we are going to provide the reader with some important machine learning concepts and techniques, that have been used in this thesis. Machine learning emerged as a subfield of computer science [3], from the study of pattern recognition and artificial intelligence. In finding an accurate description of machine learning, we refer to Arthur Samuel (1959): *"Field of study that gives computers the ability to learn without being explicitly programmed"* [24]. Machine learning deals with the study and creation of algorithms, that learn from and make predictions on data. It enables to make data-driven predictions or decisions expressed as outputs, based on example inputs, without needing strictly static program instructions. To some extent, machine learning models allow researchers, data scientists and engineers to reason about decisions, and allow them to gain "hidden insights" through learning from relationships and trends within (big) data [18][25]. Machine learning and statistics are closely related fields, in the sense that some statisticians have adopted methods from machine learning and vice versa, leading to a combined field that they call statistical learning [17].

2.2.1 (Un)supervised learning

Machine learning techniques can be generally divided in the following classes: supervised learning, unsupervised learning, reinforcement learning and other hybrid approaches. As opposed to supervised, unsupervised learning is used when data is provided without targets (i.e. response variable). In this case, we can infer a function to describe hidden structure from the provided data. As a consequence of the "unlabeled" data, there is no error or reward that can be calculated. Some examples of unsupervised learning tasks are clustering (k-means clustering, mixture models, hierarchical clustering, etc.) and dimensionality reduction (PCA, FA, etc.).
Supervised and unsupervised techniques are often combined. For instance, if every observation $\mathbf{x}_i \in \mathbb{R}^p$ is defined in a high-dimensional feature space ($p >> n$), one may use unsupervised learning techniques, such as PCA, in order to project the data to a low-dimensional feature space, while retaining the most important information. Hence, these techniques can be ideally used prior to unsupervised learning, in order to eliminate the "curse of dimensionality" in machine learning [11]. Finally, supervised learning techniques can be further divided in classification or regression tasks, depending on whether the response variable is discrete or continuous. Other techniques, such as reinforcement learning, will not be further discussed.

2.2.2 Bias-variance tradeoff

When working with prediction models, one needs to be aware of the bias-variance tradeoff. For instance, if we want to predict a target variable Y_i, given input $\mathbf{x}_i$, we may assume a (normal

error) linear relationship, given as

$$Y_i = f(\mathbf{x}_i) + \epsilon_i, \tag{2.4}$$
$$\epsilon_i \sim \mathcal{N}(0, \sigma^2),$$

with ϵ_i the irreducible error term between the model and the ground truth.
We want to find a function $\hat{f}(.)$ that approximates the true relation $f(.)$, for instance by using the unbiased ordinary least squares solution, which minimizes the residual sum of squares (SSE) $(y_i - \hat{f}(\mathbf{x}_i))^2$. However, due to the irreducible error term ϵ_i, we will not expect to find a perfect approximation. Moreover, it turns out that expected error evaluates as

$$E[(Y_i - \hat{f}(\mathbf{X}_i))^2] = \text{Bias}[\hat{f}(\mathbf{X}_i)]^2 + \text{Var}[\hat{f}(\mathbf{X}_i)] + \sigma^2, \tag{2.5}$$

where

$$\text{Bias}[\hat{f}(\mathbf{X}_i)] = E[\hat{f}(\mathbf{X}_i)] - f(\mathbf{X}_i), \tag{2.6}$$

and

$$\text{Var}[\hat{f}(\mathbf{x})] = E[(\hat{f}(\mathbf{X}_i) - E[\hat{f}(\mathbf{X}_i)])^2]. \tag{2.7}$$

All the three terms in (2.5) are strictly positive, such that σ^2 forms a lower bound on the expected error on unseen samples. More informally, by minimizing our cost function, we will automatically minimize two sources of error which prevents our supervised learning algorithm from generalizing on new unseen data. A high bias can cause an algorithm to miss the relevant relations between predictors/features and response/target (underfitting), whereas the variance error results from sensitivity to small fluctuations in a given dataset (i.e. training set). We say that a high variance will cause overfitting, as it learns the random noise rather than the relationship between features and target. To illustrate the effect of overfitting and underfitting, in function of some arbitrary chosen model which can be changed in terms of complexity (e.g. regularisation, number of included features etc.), we refer the reader to Figure 2.6. Increasing the complexity of the model will result in a higher variance error and hence overfitting, which can be seen as an increase in validation prediction error (i.e. on new unseen data) and decrease in training prediction error (i.e. on training data), whereas the opposite leads to underfitting, due to a too restrictive model.

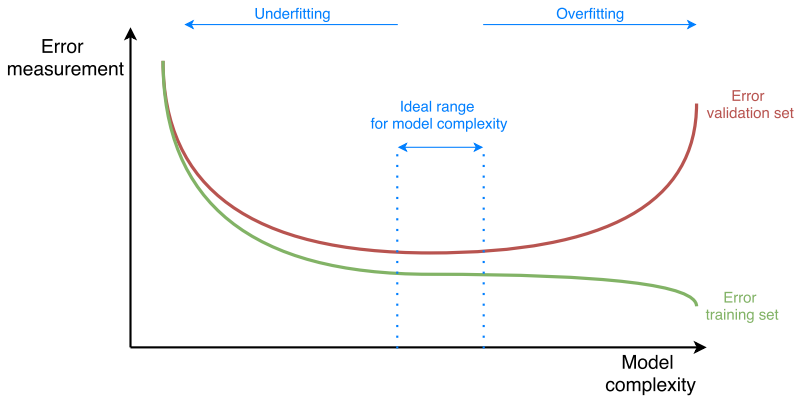


Figure 2.6: Overfitting and underfitting in machine learning models. Complex models will tend to memorize the data, without learning effectively. The latter can be seen by a decreasing training error and increasing validation error.

One can always find a good compromise between bias and variance, by means of various regularisation techniques. For instance, when it comes to artificial neural networks, different regularisation techniques currently exist, such as real-time data augmentation, dropout, L_1 and/or L_2-norm loss contraints, etc.

2.2.3 Artificial neural networks

Now that we have briefly discussed important machine learning concepts, we will move on to the even more important part on (convolutional) neural networks. Recalling the introductory chapter (see Section 1.3), we will focus on deep learning data-driven models, or more particular, deep artifical neural networks. Inspired by biological neural networks (i.e. central nervous systems of animals), artificial neural networks form a family of models, used in machine learning and cognitive science. Similar to other machine learning models, neural networks have been used to solve a wide variety of tasks, like computer vision and speech recognition, that are hard to solve using ordinary rule-based programming.

2.2.3.1 Feedforward neural networks

As seen in central nervous systems, neural networks mainly consist of interconnected neurons whose activations define recognizable linear pathways. Using axon terminals, connected via synapses to dendrites on other neurons, signal or message passing can occur between different neurons. As the sum of the input signals in some neuron exceeds a certain threshold, the neuron transmits an electrical signal along the axon. Similar to biological neural networks, feedforward neural networks are also presented as fully connected (neural) layers, each consisting of an arbitrary number of artificial neurons. With fully connected we mean that every neuron is connected to every neuron in the next layer. For instance, let us define a neural network with

an input layer of D inputs, followed by one layer, consisting of M neurons and an output layer of K outputs. We denote the input variables as $x_1, \ldots, x_D$. To give the reader a more general idea of the input, we could see the input variables as all pixel values of some global temperature map $\mathbf{x} = \{x_1, \ldots, x_D\}$. We start with M linear combinations of the input variables in the form

$$a_j = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}, \tag{2.8}$$
$$j = 1, \ldots, M.$$

With the superscript $^{(1)}$ we denote that the corresponding parameters are in the first layer of the network. In practice, one talks about *weights* for $w_{ji}^{(1)}$, and *biases* for $w_{j0}^{(1)}$. The resulting linear combinations a_j are often referred to as *activations*.
The latter activations are then transformed by using a differentiable, nonlinear *activation function* $h(.)$ to give $z_j = h(a_j)$. In literature, z_j is also called *hidden unit* (i.e. the output of a neuron). The hidden unit values are then passed to the subsequent layer where we iteratively calculate

$$a_k = \sum_{i=1}^{M} w_{ki}^{(2)} z_j + w_{k0}^{(2)}, \tag{2.9}$$
$$k = 1, \ldots, K,$$

with K the total number of outputs. As for now, we have calculated the transformation for the second layer of the feedforward neural network. Finally, the output unit activations are transformed using a chosen activation function to give a set of network outputs y_k. The corresponding network diagram can be seen in Figure 2.7. When the architecture of an arbitrary neural network is defined, we then train the corresponding model by "feeding" inputs to the network, whereafter weights and biases are updated by means of an optimization algorithm such as stochastic gradient descent. The gradient is obtained by applying the backpropagation algorithm [19].

2.2.3.2 Convolutional neural networks

Inspired by the biological visual cortex, as seen in many organisms including humans, convolutional neural networks (or ConvNets, CNNs) are feature extraction models which are capable of visual pattern recognition, in data such as video or imagery [5]. Considering the visual cortex, being the most powerful visual processing system in existence, it seems reasonable to emulate its behaviour. CNNs try to achieve this behaviour by using different building blocks in a hierarchical layer structure, consisting of convolutional layers and pooling layers. In a convolutional layer, two-dimensional filters (or kernels) are convolved in a discrete way, across the width and height of a given input. For complex-valued functions f and g, defined on $\mathbb{Z}$, the

one-dimensional discrete convolution of f and g is defined as

$$(f * g)[n] = \sum_{u=-\infty}^{\infty} f[u]g[n-u]. \tag{2.10}$$

For f, g defined on $\mathbb{Z}^2$, we find

$$(f * g)[m,n] = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} f[u,v]g[m-u, n-v]. \tag{2.11}$$

Assume that we have an input $\mathbf{x}$ with size $H \times W \times C$, where H, W are height, width and C denotes the number of channels (e.g. c_1 might be considered as the R channel of an RGB image). A filter, in a 2D convolutional layer, for feature map k and channel c, is determined by $\mathbf{w}^{kc}$ and $\mathbf{b}^{kc}$. Note that every filter that is convolved across the input results in a feature map. A convolutional layer can include several feature maps (i.e. a filter bank) and hence the number of filters are hyperparameters of a CNN. The obtained filter bank, for the above-mentioned input, is then given as

$$\mathbf{z}^{(k)}[x,y] = h\left(\sum_{c=1}^{C}\sum_{x'=1}^{H}\sum_{y'=1}^{W} \mathbf{w}^{(kc)}[x',y']\mathbf{x}[x-x', y-y', c] + \mathbf{b}^{(kc)}\right), \tag{2.12}$$

with $h(.)$ a chosen activation function. The obtained equation (2.12) can also be extended for 3D convolutional layers (e.g. when spatio-temporal data is used as input). After calculating the activations $\mathbf{z}^{(k)}$, we can then use this as new input for the next convolutional layer. The new input would then yield size $H \times W \times K$, with K channels.

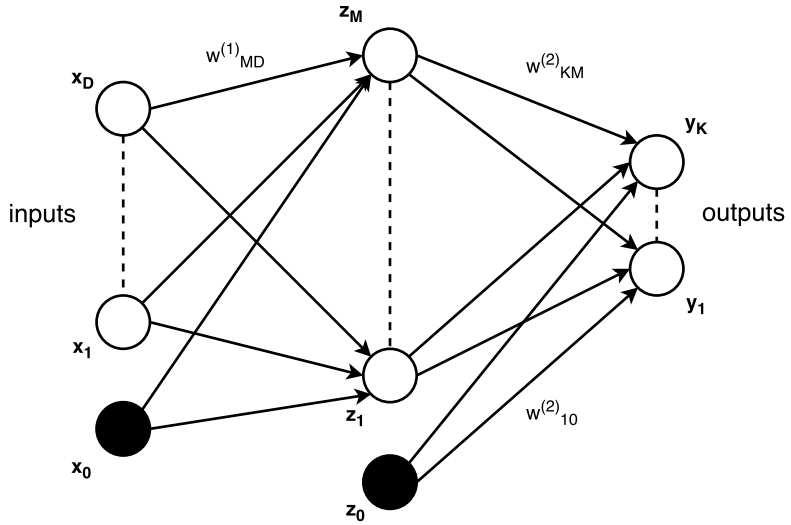
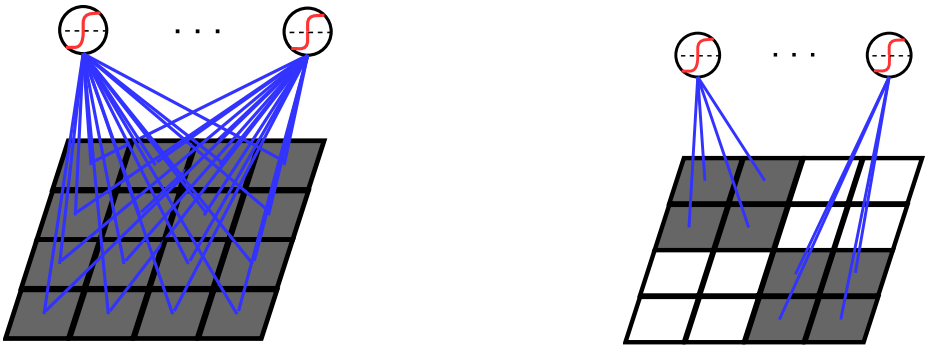


Figure 2.7: Diagram for a two-layer feedforward neural network. The input, hidden and output variables are represented by nodes, while the weight parameters are shown as links between the nodes. Bias parameters are denoted by links coming from black nodes [4].



(a) Full connectivity *(b) Local connectivity*

Figure 2.8: Difference between local connectivity in CNNs and full connectivity in ANNs.

In general, convolutional layers result in less parameters, in contrast to fully connected dense layers (recall Section 2.2.3.1). The local connectivity property in CNNs (which is similar to receptive fields within the visual cortex) are depicted in Figure 2.8b, while Figure 2.8a shows the fully connectivity in dense layers. The two hidden units in Figure 2.8b can be seen as two partial results of one filter (with some parameters) that is convoled across the input. For instance, when filters of size 10×10 for an input of 100×100 are used, one would need 100 parameters in contrast to 10000 parameters for a fully connected dense layer. Therefore, CNNs are ideally used for spatial/spatio-temporal data, where the number of inputs can be significant large, and where spatial structure is present in the data, which can be exploited.

2.2.3.3 Data-driven modeling with CNNs

As the reader recalls the introductory chapter, where we discussed the problems with pixel-wise models and the need for exploiting neighbouring information, CNNs might hence be potential candidates. Indeed, using convolutional layers, spatio-temporal features can be extracted in order to predict vegetation. A second rationale is given by the reduced complexity in convolutional layers, which then allow for global scale learning (i.e. global feature maps can be used as inputs). In terms of the extended models, we can also reserve different channels for different feature maps, on which a different and sufficient number of filters can be convoled, which then allows for deep feature extraction. After learning spatio-temporal deep features, fully dense layers could be used in order to predict some future state of vegetation. Note that, due to the spatio-temporal nature of our dataset, one would need 3D convolutional neural networks or complex recurrent neural networks (e.g. long short-term memory or LSTMs) in the end layers, which then allows to capture temporal information.

2.3 Non-linear Granger causality framework

Recalling Figure 2.1, we have now discussed the two first modules of the general framework for climate change attribution research. Climate data is split into different feature maps defined over time, on which a baseline model and different extended models can be trained and validated. In Chapter 3, we will further discuss the general implementations, strategies and corresponding results in detail. We end this chapter by discussing the last module, which will collect output from the data-driven modeling step, together with the raw/residual climate data, and on which various statistical inference techniques will be applied. We have already briefly discussed the general approach for causal inference, based on the (predictive modeling) Granger causality test. That is, comparing the predictive performance of a baseline model f_b (i.e. a model with lagged values of vegetation on future state vegetation) and various extended models $f_{e_1}, \ldots, f_{e_k}$, which are nested models with different features included. The null hypothesis of the Granger non-causality was then formulated as the null hypothesis that baseline f_b and all extended models $f_{e_1}, \ldots, f_{e_k}$ would yield the same prediction error, whereas the alternative would be that each extended model predicts significantly better than baseline. The original formulation of the Granger causality test is only defined for linear models, hence making it not applicable for highly non-linear models such as CNNs. In this thesis, we've decided to work with an adapted non-linear Granger causality approach which is highly related to the linear approach [23]. That is, baseline and extended models can be replaced by non-linear models and predictions are then analogously compared. Moreover, in order to make statistical relevant conclusions on the obtained predictions, a (nonparametric) Mann-Whitney U test is used in combination with bootstrapping.

Formally, assume that we have trained two CNNs, which correspond to the baseline model f_b and some extended model f_e respectively. The extended model includes one feature of interest, next to vegetation. Assume that both models were seperately trained on the same data, with the same network parameters/configuration (e.g. number of layers, filters, hidden units, activation functions etc.), but however with different number of channels. That is, the baseline model would only use one channel in its input layer, whereas the extended model would use two: one for vegetation and the other for the feature of interest. Hence, we assume that both models are comparable. After training, we then separately test both models on a well-defined (test) set. We emphasize that the latter set should be non-overlapping with the training set. This could be obtained by dividing the available time frame in two separated chunks, with an additional *gap window* (i.e. time frame which is unused) introduced, in order to make both sets as different as possible. The obtained predictions on the test set are denoted by $\hat{f}_b(t)$ for the baseline model and $\hat{f}_e(t)$ for extended model respectively, where t denotes the timestamp of some arbitrary prediction. The reader should already notice that each prediction represents some future vegetation state. Further assuming that we've corresponding vegetation targets $Y_v(t) \in \mathbb{R}^n$

available, where n denotes the number of pixels of the vegetation map, we can hence calculate the squared prediction errors as

$$\tilde{Y}_b(t) = \left(Y_v(t) - \hat{f}_b(t)\right)^2, \qquad (2.13)$$
$$\tilde{Y}_e(t) = \left(Y_v(t) - \hat{f}_e(t)\right)^2.$$

Taking in account the obtained squared prediction errors, we could now easily calculate and compare the mean squared prediction errors for both models. However, as we are interested in statistical relevant conclusions, a well-defined hypothesis test would be more appropriate. Hence, a hypothesis test could be conducted by testing the null hypothesis that an arbitrary chosen prediction error for the baseline model will most likely be smaller or equal, compared to a prediction error of the extended model. The nonparametric Mann-Whitney U test could be ideally used, as it doesn't make any assumption about the underlying distribution of the two different prediction errors [22]. However, taking in account one of the assumptions of the latter test, samples from both groups need to be independent. As samples (i.e. squared prediction errors) from one group (e.g. baseline model) are time dependent, it follows that the independence assumption would not be fulfilled, hence using the Mann-Whitney U test would not be appropriate [10]. However, the latter violation of independence could be (partially) reduced by using bootstrapping techniques, in combination with the Mann-Whitney U test. That is, by obtaining an arbitrary amount of bootstrap samples for each group, both bootstrap distributions can be used as an approximation of the true distribution of squared prediction errors. Finally, the Mann-Whitney U test is used in order to test the null hypothesis that the central location of the squared prediction errors (i.e. bootstrap distribution) for the baseline model is smaller, compared to the extended model. When the included feature of the extended model G-causes vegetation, we would hence expect that the null hypothesis would be rejected in favour of the alternative. For more details on implementations and strategies, concerning the non-linear Granger causality framework, we refer the reader to Chapter 3.

3 Results

In this chapter we move on to a more concise discussion of data preprocessing, model building and corresponding results, obtained throughout this thesis. In Chapter 2, we have introduced the reader to the available datasets and general methodology, corresponding to our proposed framework for data-driven climate change attribution. Prior to discussing the used models and architectures (see Section 3.2), we first start with data preprocessing (see Section 3.1). Finally, we provide the reader with the obtained results and corresponding discussions (see Section 3.3). For a final and brief summary on conclusion and future work, we refer the reader to Chapter 4.

3.1 Data preprocessing

Recalling Section 2.1, we will use the Leaf Area Index dataset (GLASS) for the outcome of interest (i.e. vegetation), together with the temperature (ERA) and precipitation (MSWEP) datasets, representing the features/predictors. In contrast to the Granger causality test, typically applied on individual time series, we will work with global input and output maps (i.e. images). Due to the difference in temporal coverage, as well as temporal resolution (see Table 2.1), between each dataset, we choose to work with a coverage starting from 1982 and ending at 2008, at a temporal resolution of nine days. As the reader already noticed, due to the choice of nine days temporal resolution, limited information would be used from the temperature and precipitation dataset. Hence, rather than extracting raw data at each 9th timestamp, samples in-between two consecutive 9th timestamps will be interpolated by means of calculating the average across the eight samples in each interval, resulting in a sequence of feature maps $\mathbf{X}_n$

with corresponding target (LAI) maps $\mathbf{Y}_n$. Continuously, each sequence of feature maps $\mathbf{X}_n$, as well as target (LAI) maps $\mathbf{Y}_n$, are scaled as

$$\begin{aligned}\mathbf{X}_n^s &= \frac{\mathbf{X}_n - \mathbf{X}_n^-}{\mathbf{X}_n^+ - \mathbf{X}_n^-},\\ \mathbf{Y}_n^s &= \frac{\mathbf{Y}_n - \mathbf{Y}_n^-}{\mathbf{Y}_n^+ - \mathbf{Y}_n^-},\end{aligned} \tag{3.1}$$

resulting in a $[0, 1]$ scale, where the superscripts $+$ and $-$ denote the maximum and minimum over the (global) sequences, respectively. From now on, we will ignore the subscript n and superscript s. Instead, we will use subscripts t, p, to denote the particular feature map sequence (i.e. temperature, and precipitation) of interest. Other feature maps might be considered, but in this thesis, we will focus on temperature and precipitation. Moreover, since we have silently ignored the residual datasets (see previous chapter, Section 2.1.2), we will use the superscript R to denote residual feature/target map sequences. In order to deal with missing values/pixels (e.g. for LAI or precipitation), which denote "empty" locations such as sea or lake locations, we will replace corresponding N/A values with -1.

Finally, we move on to a final data preprocessing step, taking in account and allowing for future model building and selection. Informally, we will partition the feature and target map sequence data in a training and test set. However, due to the sequential nature of our dataset, techniques such as K-fold cross-validation, are less/not applicable to our training set, due to the correlation between train and test folds. Hence, we will look at a more sequential approach for obtaining reliable out-of-sample performance measurements, while controlling on generalisation, overfitting and underfitting.

The proposed training and validation approach is illustrated in Figure 3.1. Each (time) sequence of datapoints (i.e. the initial training set) is similarly split in a training and validation set, where a sufficiently large *gap window* is introduced, in order to avoid correlation between training and validation samples. For both the training and the validation set, we then extract samples $\mathbf{S}_i$ as

$$\mathbf{S}_i := \Big(\mathbf{Y}(i : i + T), \mathbf{X}_{t,p}(i : i + T), \mathbf{Y}(i + T + L)\Big), \quad \forall i \in \{0 \dots\}, \tag{3.2}$$

where T denotes the time window length and L the lag, respectively. Note that the above formulation corresponds to the Granger causality test, as defined in (1.3). Indeed, we are interested in predicting some future state of vegetation $\mathbf{Y}(i + T + L)$, by using lagged values for vegetation $\mathbf{Y}(i : i + T)$, as well as lagged values for temperature and precipitation $\mathbf{X}_{t,p}(i : i + T)$. Furthermore, in traditional Granger causality analysis, a one-step-ahead forecast is considered, which corresponds to the case where $L = 1$. Take in mind that here we denote the lag L as a time window, defined between past states and future state, in order to avoid strong similarity between the two states (e.g. recall the non-stationary and cyclic behaviour of the raw data, as discussed in Section 2.1). Moreover, here we use the term *time window* T, to denote the number

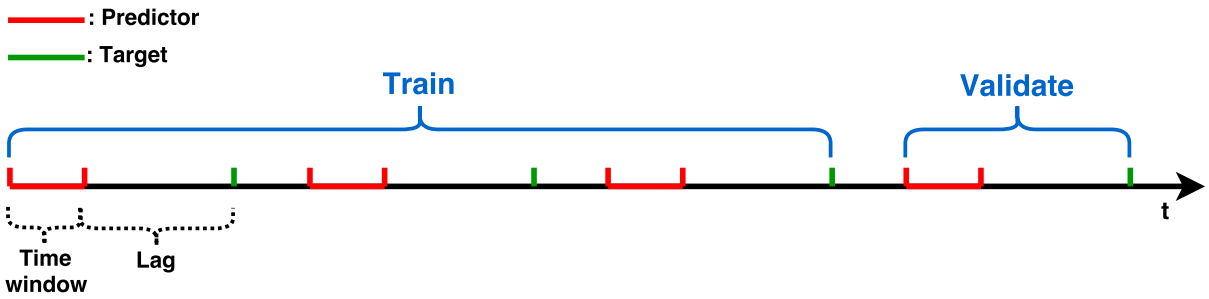


Figure 3.1: General approach for the extraction of training and validation samples, where the time axis represents a subset of the feature/target map sequence. Note that the other subset, which denotes the final test set, is not shown here.

of past states. In general, we will train some arbitrary chosen model on the training samples and validate by using the validation samples. After training and validation, we then use the test set, in order to obtain final out-of-sample prediction errors. The extraction of samples from the test set is analogously explained as for the training and validation set, and again, a sufficiently large gap window is used between the validation and the final test set. Note that there exist other approaches to evaluate the out-of-sample accuracy of forecasting models, however, the conservative methodology outlined above was chosen because it rules out over-optimistic model performance due to correlation between training and test data. In conclusion, we refer the reader to Table 3.1, for an overview of the sizes corresponding to the training, validation, test and gap set. The latter sets will be used for the subsequent model building and selection. Note that the total number of samples ($n = 1218$) is relatively small, due to the temporal resolution of nine days. Besides that, due to the high complexity, which comes with the machine learning models of interest, as well as limited computational resources, we are forced to use a limited training set.

Dataset	**Coverage**	**Coverage (year)**	**Samples** (n)
Training	40%	1982-1992	487
Validation	25%	1994-2000	304
Test	25%	2002-2008	307
Gap	10%	-	120
Total	-	1982-2008	1218

Table 3.1: General overview of the sizes for each type of set, used for model building and selection.

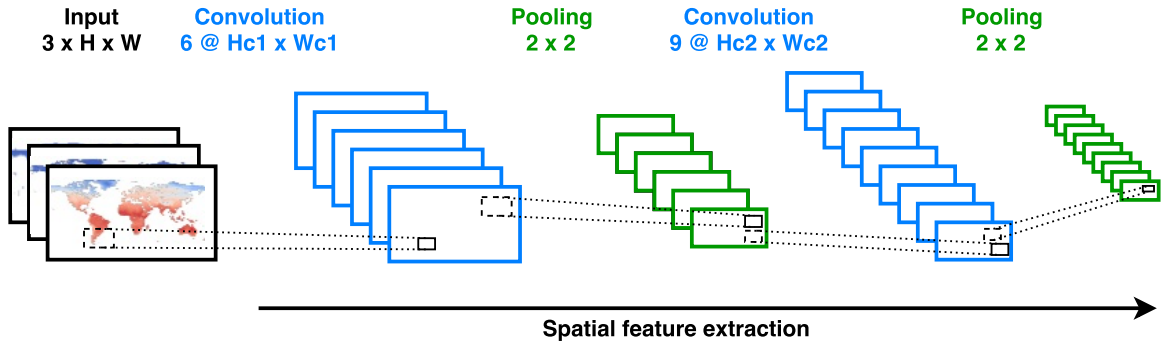


Figure 3.2: Example of a simple convolutional network, consisting of an input layer with size $(3 \times H \times W)$ and two convolutional-pooling layers. The input is obtained, as defined in (3.2).

3.2 Model building

After obtaining the preprocessed data, we are now ready to build different model architectures, which can be used for model training, validation and testing. The models will be trained on the data as seen in Table 3.1. Although different architectures were studied in this thesis, in conclusion we will focus on one particular model (i.e. temporal convolutional networks, Section 3.2.3), together with the corresponding model evaluations (see Section 3.3). The rationale for using this model is explained by relative low runtime and memory complexity. Nevertheless, we will briefly discuss more complex architectures, which can be used in high performance computing (HPC) settings. Note that from now on, we will omit the prefix word "neural" in "neural networks", as each discussed model is in fact a special neural network model. Finally, we refer the reader to Appendix A, for more information on the used software and hardware specifications.

3.2.1 Convolutional/locally recurrent networks

Recalling Section 2.2.3.2, the reader was introduced to convolutional neural networks (CNNs), together with the rationale for using the latter models as potential data-driven models in our proposed framework (see Figure 2.1). Indeed, convolutional layers can be used to exploit neighbouring information, by means of learning spatial features, as well as for global scale learning, due to its reduced complexity, in contrast to fully dense layers in feedforward neural networks. In Figure 3.2, a simple convolutional network is illustrated. Here, the input is given as a 3-channel image or sample, where the channels correspond to a vegetation, precipitation and temperature feature map $\mathbf{Y}(t), \mathbf{X}_{p,t}(t)$, evaluated at some arbitrary timestamp t. Each channel/map has a size $H \times W$, with H and W as height and width, respectively. The latter input is fed to a convolutional layer, consisting of six different filters with size $H_{c1} \times W_{c1}$. The filters are then convolved with the input, resulting in six different output maps. Then, the latter output maps are fed to a (second) pooling layer (e.g. max pooling, min pooling etc.), where each map is reduced with factor two in both dimensions. The previous steps are then similarly

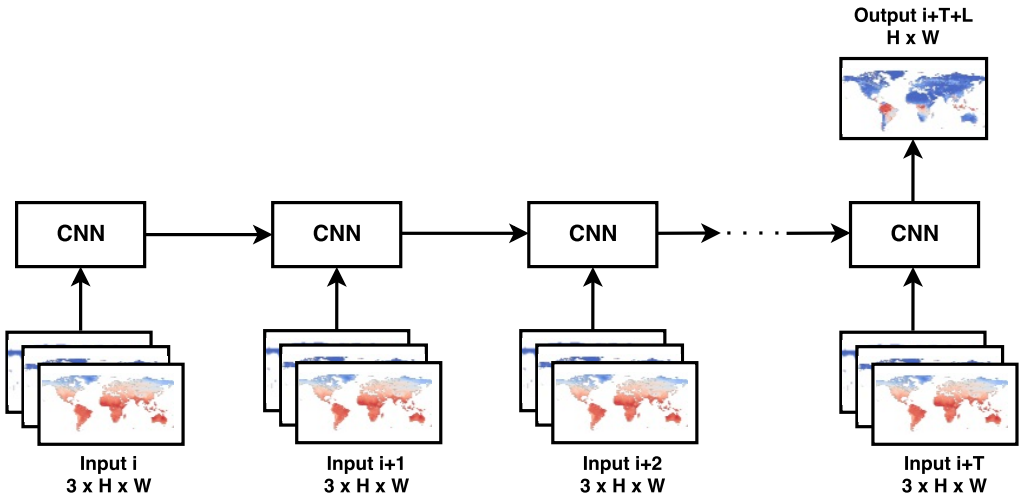


Figure 3.3: Example of a convolutional recurrent network, with input and output defined in (3.2). Note that the network is unfolded in time, which allows for training by means of the backpropagation through time algorithm [27]. However, as T increases, training becomes computationally intensive.

repeated for the consecutive two layers, where the final convolutional layer now consists of nine different filters, with size $H_{c2} \times W_{c2}$. In practice, convolutional networks can consist of many subsequent convolutional-pooling layers, in order to obtain (deep) spatial features at the end layer. The latter features are then used as input to a second network (e.g. fully dense, recurrent, etc.), for further processing. Moreover, in the context of this thesis, we can use convolutional layers within a recurrent neural network (i.e. convolutional recurrent networks), allowing for spatio-temporal learning, as illustrated in Figure 3.3. Informally, this model is sequentially trained (and validated) by using samples $\mathbf{S}_i$, as defined in (3.2), where the relationship between input $\mathbf{Y}(i:i+T), \mathbf{X}_{t,p}(i:i+T)$ and output $\mathbf{Y}(i+T+L)$ is learned. More formally, outputs h_t are defined as

$$h_t = \sigma(f_i(x_t) + f_h(h_{t-1})), \tag{3.3}$$

with $\sigma(.)$ denoting an activation function of interest, $f_i(x_t)$ the output of the convolutional neural network for input x_t and $f_h(h_{t-1})$ the new hidden state or output of the convolutional neural network, calculated for the previous hidden state h_{t-1}. The model is unfolded in time, which then allows for training, by means of the backpropagation through time (BPTT) algorithm [27]. For increasing T, the latter algorithm becomes memory intensive, although computational efficient methods currently exist [14]. However, taking in account the temporal coverage for the dataset, we can use a smaller time window T. Note that each CNN block, in Figure 3.3, represents a convolutional network (i.e. as seen in Figure 3.2), extended with a fully dense network. This network is fed with the spatial (deep) features, obtained by the convolutional network, in order to reconstruct the target vegetation map. Due to the fully connectivity in the latter network, we will only use one dense layer with $180 \times 360 (= 64800)$ hidden units, representing the

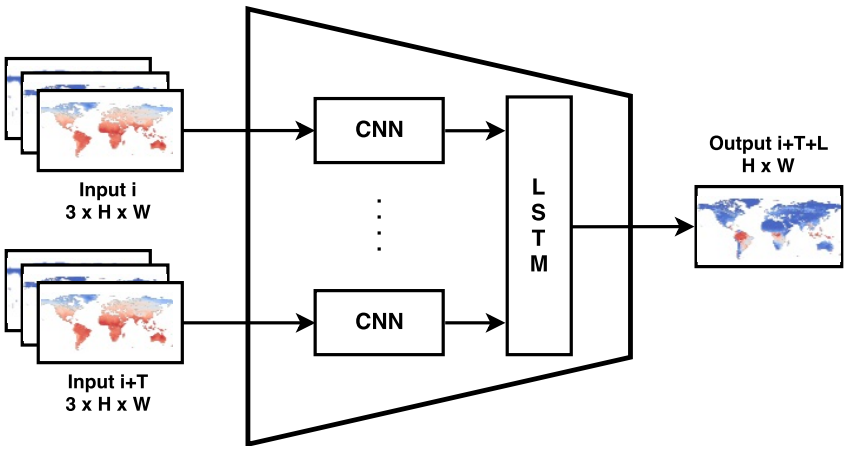


Figure 3.4: Example of a convolutional LSTM network, with input and output defined in (3.2). The LSTM block represents the recurrent part of the network, which is analogously structured as in Figure 3.3. Note that in this case, CNN blocks are replaced by LSTM blocks.

target vegetation map pixels. Hence, the fully dense network represents the output layer of the convolutional neural network. In general, this layer is also called the *bottleneck layer*, as the total amount of network (hyper)parameters is dominated by the number of parameters (=64800) in this layer. Finally, we can also use locally connected layers in the convolutional network, where the parameter sharing is eliminated and hence resulting in distinct filters for each input location. Hence, we can argue that locally connected layers are more able to retain and capture local properties within the feature maps, in contrast to convolutional layers. However, this will come at the expense of increasing number of parameters, and hence, memory complexity. Alternatively, we can also increase the number of filters in each convolutional layer, in order to approximately emulate the behaviour of locally connected layers.

3.2.2 Convolutional LSTM networks

When temporal dependencies (e.g. between vegetation and/or features) exist over a large time domain, standard recurrent neural networks (e.g. convolutional recurrent networks) would fail, due to restrictive memory. This issue was adressed by Hochreiter & Schmidhuber (1997), who also proposed (at the time of writing) Long Short Term Memory (LSTM) networks, capable of learning long-term dependencies [16]. Over the past decade, this type of network has proven to work tremendously well on a large variety of problems, such as speech recognition, language modeling, image captioning, etc. Informally, the latter models are able to memorise information, during long periods of time. The architecture for a convolutional LSTM network is shown in Figure 3.4. Note that the LSTM block can be illustrated as seen in Figure 3.3, where CNN

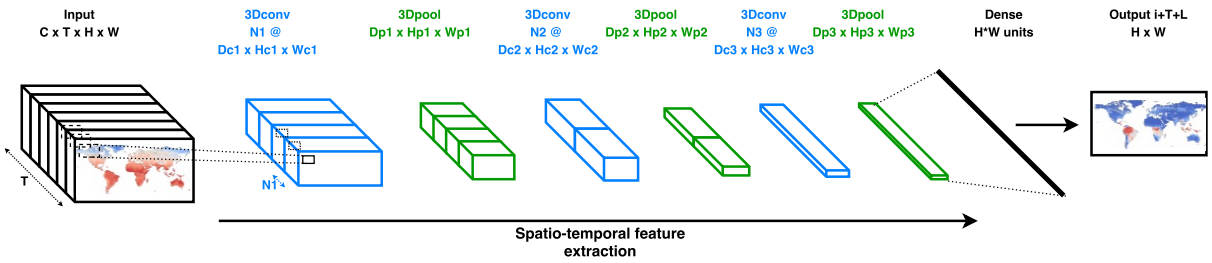


Figure 3.5: Example of a temporal convolutional network. Note the difference between the previous convolutional networks, e.g. as illustrated in Figure 3.2. In this case, the (convolution/pooling) filters are defined in three dimensions, as opposed to the two-dimensional filters in the previous networks.

blocks are replaced with LSTM blocks. Furthermore, each LSTM block is defined as

$$\begin{aligned} i_t &= \sigma_i(x_t W_{xi} + h_{t-1} W_{hi} + w_{ci} \circ c_{t-1} + b_i), \\ f_t &= \sigma_f(x_t W_{xf} + h_{t-1} W_{hf} + w_{cf} \circ c_{t-1} + b_f), \\ c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(x_t W_{xc} + h_{t-1} W_{hc} + b_c), \\ o_t &= \sigma_o(x_t W_{xo} + h_{t-1} W_{ho} + w_{co} \circ c_t + b_o), \\ h_t &= o_t \circ \sigma_h(c_t), \end{aligned} \tag{3.4}$$

with $\circ$ the Hadamard product, i_t, f_t, c_t, o_t and h_t, the input gate, forget gate, cell state and hidden state, respectively [13]. Note that all the weights W_x, W_h in an LSTM block are used to direct the operation of the gates. That is, the latter weights occur between the values that feed into the block (i.e. the input vector x_t, and the previous hidden state h_{t-1}) and each of the gates. Hence, the LSTM block learns to decide how to control its memory, as a function of the latter values. Again, LSTM blocks are trained by means of the BPTT algorithm.

3.2.3 Temporal convolutional networks

Finally, we propose our last model, which differs from the aforementioned models within the context of temporal learning. Moreover, for the previous models, a recurrent part was introduced, next to the convolutional part. Both parts were connected, hence allowing for spatio-temporal learning. Although the convolutional LSTM network could be ideally used for capturing long-term dependencies, we emphasize that the latter recurrent networks come with high computational complexity. Besides that, taking in account the temporal coverage of nine days, as well as the limited size of our dataset (see Table 3.1), the latter models would result in low generalisation performance. Indeed, complex models with many (hyper)parameters, will tend to overfit on a limited dataset, whereas a limited number of available training examples would restrict models from learning effectively (i.e. underfitting). Consequently, in our last model, we will eliminate the recurrent part and instead extend the convolutional network, allowing for

spatio-temporal learning. The proposed architecture, corresponding to this model, is illustrated in Figure 3.5. Each input has size $C \times T \times H \times W$, with C the number of channels or input features (vegetation incl.), T the time window size and H, W the height and width for each input feature and vegetation map, respectively. The number of cuboids for the input, convolutional and pooling layers, corresponds to the size of the time window. Hence, for this particular architecture, the time window is reduced from size $T(= 6)$ to size one, by means of three consecutive convolutional-pooling operations. In contrast to the aforementioned convolutional network (see Figure 3.2), convolution and pooling filters are now defined in three dimensions, where the first dimension corresponds to the depth over the time domain, and the last two to the spatial domain, respectively. By increasing the hyperparameters N_1, N_2 and N_3, for the three convolutional layers, more distinct filters are learned and convolved with the corresponding inputs. Finally, the output of the third pooling layer is flattened and fed to a fully dense network part, consisting of one (output) layer with $H \times W$ hidden units. Moreover, the final output layer corresponds to the predicted future state of vegetation $\mathbf{Y}(i + T + L)$, given the i-th input $\mathbf{Y}(i : i + T), \mathbf{X}_{p,t}(i : i + T)$.

3.3 Model selection and discussion

Up to now, we have proposed different data-driven models, for the prediction of future states of vegetation $\mathbf{Y}(i + T + L)$, given T past states of vegetation $\mathbf{Y}(i : i + T)$ and features $\mathbf{X}_{p,t}(i : i + T)$. In this section, we discuss the training, validation and test results, together with more in-depth model evaluations. Moreover, as already discussed in Section 3.2.3, we will restrict to the discussion of temporal convolutional networks only.

3.3.1 Architecture and (hyper)parameters

The proposed architecture for the temporal convolutional network, obtained after extensive model training and tuning, is shown in Table 3.2. Recall the general discussion, for this architecture, in Section 3.2.3. Again, the rationale for using one fully dense layer (i.e. output layer) after the convolutional network part, is easily explained by looking at the corresponding number of parameters for this dense layer. Indeed, each of the 64800 hidden units in the dense layer (nr. 9), are connected to each output (unit) of the reshape layer (nr. 8), and hence resulting in more than 99×10^6 parameters. This particular architecture has an input, consisting of $C = 3$ feature maps (e.g. vegetation, precipitation and temperature). Although, within the context of the non-linear Granger causality framework (see Section 2.3), the number of channels will differ, depending on the type of architecture (e.g. baseline or extended). Nevertheless, the architectures for baseline and extended models will only differ in the first layer by means of the number of channels C. Hence, taking in account the temporal coverage of nine days, we choose to work with a fixed time window of six timesteps, which corresponds to approximately

two months. Analogously, the lag window is fixed at three timesteps ($L = 3$) or approximately one month, for each model. In summary, the remaining (hyper)parameters for each model, are listed below:

- **Regularization (output layer):** L_2
  - Penalty parameter (λ): $1e-5$
- **Optimization:** Nesterov momentum
  - Learning rate : 0.1
  - Momentum: 0.9
- **Cost function:** Adapted mean squared error loss function
- **Batch size:** 16
- **Training epochs:** 50
- **Time window (T):** 6
- **Lag window (L):** 3

Finally, recalling the data preprocessing (see Section 3.1), vegetation and feature maps are dominated by "empty" locations, such as sea pixels. Logically, when it comes to predicting vegetation, the latter pixels can be ideally omitted during the calculation of the MSE for each prediction and corresponding target, by means of the adapted MSE loss function. By doing so, the calculated loss only corresponds to pixels of interest. Furthermore, we can argue that the latter technique allows for faster convergence and effective learning.

Nr.	Layer	#Filters/hid. units	Filter shape $(\mathbf{D}, \mathbf{H}, \mathbf{W})$	Stride $(\mathbf{D}, \mathbf{H}, \mathbf{W})$	Nonlinearity	Weight init.	Output shape $(\mathbf{C}, \mathbf{T}, \mathbf{H}, \mathbf{W})$	#Params
1	Input	-	-	-	-	-	$(3, 6, 180, 360)$	0
2	Conv3D	8	$(3, 6, 6)$	$(1, 3, 3)$	Rectify	Glorot unif.	$(8, 4, 59, 119)$	2600
3	MaxPool3D	-	$(1, 2, 2)$	-	-	-	$(8, 4, 29, 59)$	0
4	Conv3D	16	$(3, 5, 5)$	$(1, 1, 1)$	Rectify	Glorot unif.	$(16, 2, 25, 55)$	9616
5	MaxPool3D	-	$(1, 2, 2)$	-	-	-	$(16, 2, 12, 27)$	0
6	Conv3D	32	$(2, 4, 4)$	$(1, 3, 3)$	Rectify	Glorot unif.	$(32, 1, 9, 24)$	16416
7	MaxPool3D	-	$(1, 2, 2)$	-	-	-	$(32, 1, 4, 12)$	0
8	Reshape	-	-	-	-	-	(1536)	0
9	Dense	64800	-	-	Tanh	Glorot unif.	(64800)	99597600
10	Reshape	-	-	-	-	-	$(180, 360)$	0
								Total: 99626232

Table 3.2: A comprehensive tabulation of the proposed temporal convolutional network. For the corresponding visual representation, we refer to Figure 3.5. Note the significant bottleneck layer (nr. 9), representing 99% of the total parameters. The filter and output shapes are explained in Section 3.2.3.

3.3.2 Evaluation

We can now evaluate the temporal convolutional network, with corresponding architecture and (hyper)parameters (see Section 3.3.1), by means of training, validation and testing on the pre-processed datasets (see Section 3.1). Within the context of the non-linear Granger causality framework (see Section 2.3), we will first discuss the obtained training, validation and test results, for different baseline and extended models. To some extent, this allows us to reason about the different models, in terms of G-causal relationships. Additionally, more in-depth evaluations will be considered, in terms of the obtained MSE measurements for each model, as well as statistical inference by means of the MWU test in combination with bootstrapping techniques (BMWU), as seen in Section 2.3. Again, we want to emphasize that no focus will be put on drawing conclusions on causal relationships between different climate variables.

3.3.2.1 Training, validation and test results

In Table 3.3, the obtained training, validation and test errors are summarised for two different baseline models (i.e. in terms of residual or raw vegetation target maps), three different extended models with one feature and two different extended models with two features, respectively. For each model, the included predictors (i.e. vegetation and/or features) are mentioned, by using the notation as defined in Section 3.1. Further, note that the reader may assume target residual vegetation, when residual vegetation is used as predictor, and the other way round. We observe that the validation and test errors for each model do not significantly differ. The overall best performance is observed for the extended model with precipitation, hence indicating that in terms of G-causality, precipitation might have more influence on vegetation, in contrast to temperature. Also note a significant difference in test errors, between extended model with predictors $\mathbf{Y}, \mathbf{X}_p, \mathbf{X}_t^R$, and extended model with $\mathbf{Y}, \mathbf{X}_p, \mathbf{X}_t$. Moreover, the inclusion of the raw temperature feature $\mathbf{X}_t$ results in lower prediction performance, compared to the baseline model. The latter issue might be explained after a close inspection of a corresponding feature map, illustrated in Figure 2.2. Indeed, in contrast to precipitation, a more cyclic and similar behaviour as vegetation is observed for the raw temperature data. Consequently, we may argue that features, characterised by strong cyclic and similar behaviour to vegetation, are less informative for the prediction of future vegetation states. At the other hand, inclusion of the residual temperature feature $\mathbf{X}_t^R$ results in better predictions, compared to the baseline model. Nevertheless, when comparing the extended model with predictors $\mathbf{Y}, \mathbf{X}_t^R$ and extended model with $\mathbf{Y}, \mathbf{X}_p$, we still observe that precipitation is more informative. Finally, we note that models with residual vegetation data result in much lower R^2 test performance, in contrast to models with raw vegetation data. A possible explanation is given by the nature of the temporal convolutional network architecture, where the extraction of useful spatial features depends on the presence of visual structures/blobs. However, in the subsequent MSE and BMWU analysis, we will further focus on the latter issue.

Three test predictions for the baseline and extended model with predictors $\mathbf{Y}, \mathbf{X}_p, \mathbf{X}_t^R$, are presented in Figure 3.6. Corresponding targets and timestamps are given as well. In general, it seems that both models yield predictions which are visually similar to the corresponding targets. At first sight, we don't observe a significant difference between the baseline and extended model predictions. Hence, more in-depth analysis is required to compare the different models, in terms of prediction performance. In the following sections we will further compare each model, by means of further analysing MSE performance in the spatio-temporal domain and global BMWU analysis for statistical inference.

Model (#feat.)	**Predictors**	**MSE (train)**	**MSE (validation)**	**Train duration (s)**
Baseline	$\mathbf{Y}$	0.00216	0.00270	10250
	$\mathbf{Y}^R$	0.000821	0.000937	10000
Extended (1)	$\mathbf{Y}, \mathbf{X}_p$	**0.00207**	**0.00249**	12000
	$\mathbf{Y}, \mathbf{X}_t^R$	0.00207	0.00256	15250
	$\mathbf{Y}^R, \mathbf{X}_p$	**0.000812**	**0.000940**	11750
Extended (2)	$\mathbf{Y}, \mathbf{X}_p, \mathbf{X}_t^R$	0.00210	0.00251	17500
	$\mathbf{Y}, \mathbf{X}_p, \mathbf{X}_t$	0.00231	0.00286	17000

Model (#feat.)	**Predictors**	**MSE (test)**	$\mathbf{R^2}$ **(test)**
Baseline	$\mathbf{Y}$	0.00272	0.95143
	$\mathbf{Y}^R$	0.000546	-0.05930
Extended (1)	$\mathbf{Y}, \mathbf{X}_p$	**0.00249**	**0.95551**
	$\mathbf{Y}, \mathbf{X}_t^R$	0.00256	0.95442
	$\mathbf{Y}^R, \mathbf{X}_p$	0.000543	-0.05382
Extended (2)	$\mathbf{Y}, \mathbf{X}_p, \mathbf{X}_t^R$	0.00253	0.95485
	$\mathbf{Y}, \mathbf{X}_p, \mathbf{X}_t$	0.00287	0.94884

Table 3.3: Prediction errors, for different baseline and extended models. Note that for the second column, we use the notation as defined in Section 3.1. When it comes to the raw and residual vegetation models, we observe optimal performance (denoted with bold) for the extended model with precipitation, on both the training, validation and test set. Furthermore, it seems that using the residual vegetation data yields low R^2 performances. Finally, the reader may assume target residual vegetation, when residual vegetation is used as predictor.

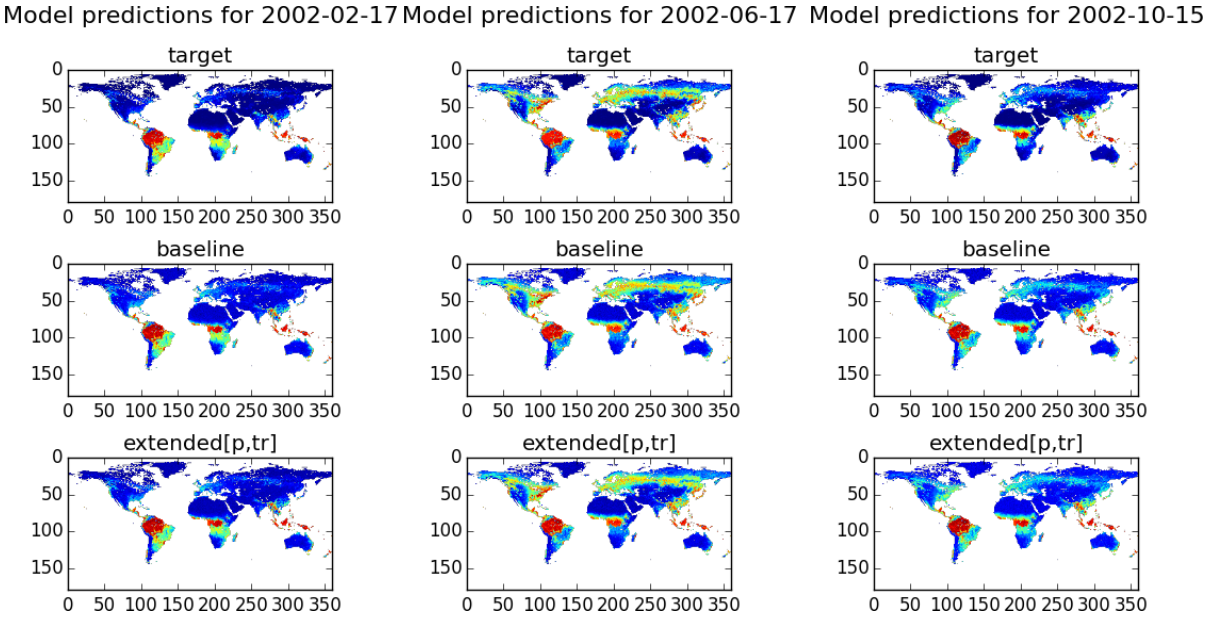


Figure 3.6: Visualisation of different (test) predictions with corresponding targets, for baseline model and extended model with precipitation and residual temperature. At first sight, the obtained predictions are similar to their corresponding target.

3.3.2.2 MSE analysis

The obtained test predictions, as well as corresponding MSE measurements for each model, can be exploited in order to allow for additional analysis. Moreover, in this section we will analyse the differences between MSE measurements for different models, on global and local (i.e. Spain) scale, evaluated over the time domain of the test set. The latter allows us to reason about baseline and extended models, in function of time, features and vegetation. For example, we refer the reader to Figure 3.7. The upper three graphs represent the average MSE differences, calculated between:

1. baseline and extended $(\mathbf{Y}, \mathbf{X}_p)$
2. extended $(\mathbf{Y}, \mathbf{X}_p)$ and extended $(\mathbf{Y}, \mathbf{X}_p, \mathbf{X}_t^R)$
3. baseline and extended $(\mathbf{Y}, \mathbf{X}_p, \mathbf{X}_t^R)$

both for global and local scale (i.e. Spain) and evaluated over the time domain of the test set (i.e. sequence of test samples). The remaining three graphs correspond to vegetation and features. When evaluating on global scale, we observe that the extended models are continuously better than baseline. However, when looking at the local evaluation plot (i.e. right plot, in Figure 3.7), we observe peak performances at each 50th timestep, together with alternate high-low performances. Additionally, we also observe that peak performances are related with vegetation peaks. The latter might be explained by the fact that features have more influence on vegetation, during time periods when vegetation is high. In terms of prediction performance, it seems that the "global scale learning" property of the temporal convolutional networks is illustrated by the optimal global performance that is observed in the global evaluation plot. Indeed, when looking at the obtained global predictions, for each model in Figure 3.8, we observe that predictions are almost identical to the targets, in contrast to local predictions (i.e. right plot, in Figure 3.8).
Similar analysis is conducted for (residual vegetation) baseline $(\mathbf{Y}^R)$ and extended $(\mathbf{Y}^R, \mathbf{X}_p)$ models, shown in Figure 3.9, 3.10. In contrast to raw vegetation models, no optimal global performance is observed. Moreover, prediction plots in Figure 3.10, indicate much lower prediction performance for both baseline and extended model, on global and local scale. Not entirely surprising, recalling the previous discussion of the training, validation and test performances for the different (residual vegetation) baseline and extended models, in Table 3.3. Finally, similar results for different local scales are provided in Appendix B.1.

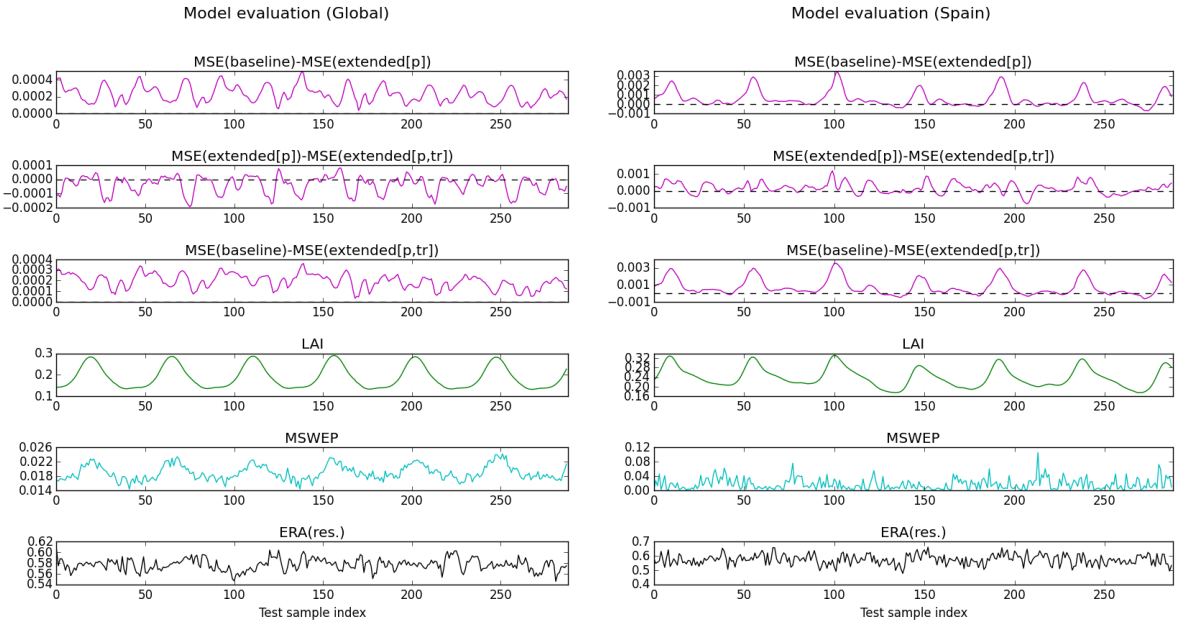


Figure 3.7: MSE model evaluation on global and local scale (i.e. Spain), evaluated over the time domain of the test set. The upper three graphs correspond to the comparison of MSE performances between two different models. Positive values indicate better performance for the first model, whereas the opposite holds for negative values . The last three graphs represent the (global/local) averages for vegetation and features, evaluated over the time domain of the test set.

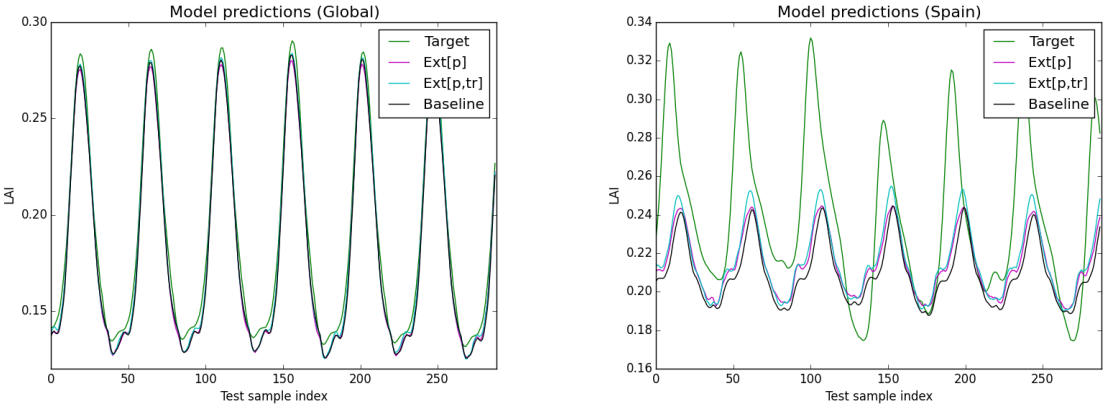


Figure 3.8: Global and local (i.e. Spain) average predictions for three different models, with corresponding targets, evaluated over the time domain of the test set.

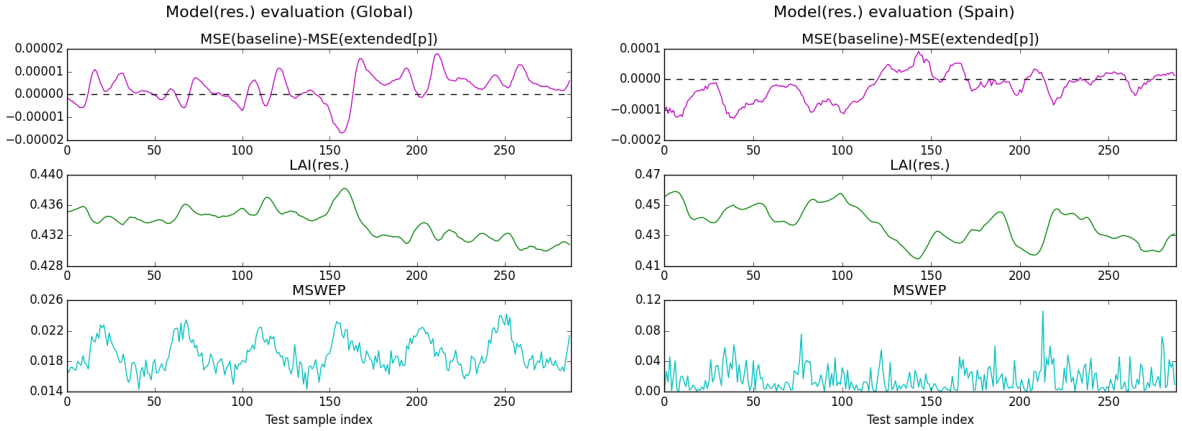


Figure 3.9: MSE model evaluation for (residual vegetation) models on global and local scale (i.e. Spain), evaluated over the time domain of the test set. The first graph corresponds to the comparison of MSE performances between baseline and extended model with precipitation. Positive values indicate better performance for baseline, whereas the opposite holds for negative values. The remaining graphs represent the (global/local) averages for residual vegetation and precipitation, evaluated over the time domain of the test set.

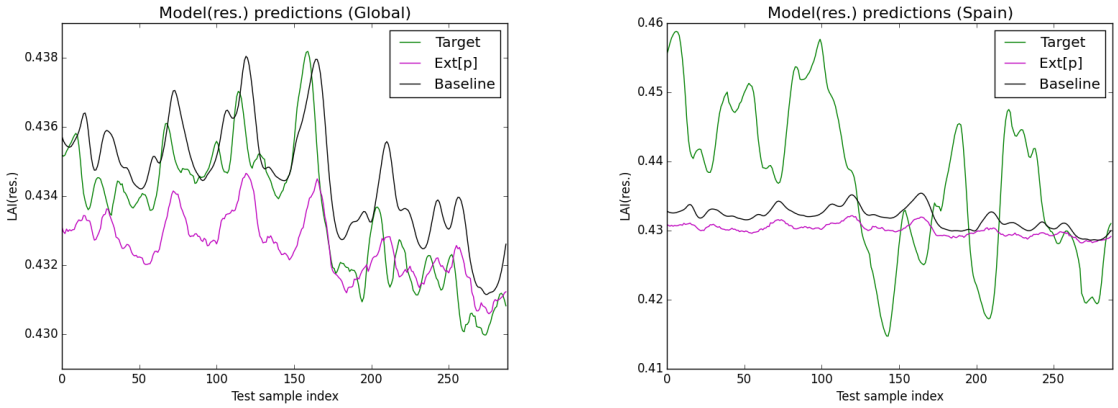


Figure 3.10: Global and local (i.e. Spain) average predictions for three different (residual vegetation) models, with corresponding targets, evaluated over the time domain of the test set.

3.3.2.3 BMWU analysis

Until now, we analysed the different models by means of looking at the corresponding training, validation and test errors, averaged across the spatial and temporal domain. Furthermore, we introduced global and local model evaluation techniques, where MSE measurements and predictions are compared between different baseline and extended models, over the time domain of the test set. As the latter allows us to study the behaviour of different models, in function of time, vegetation and features, we still need a more statistical evaluation method. Hence, in this final section we will evaluate our models by means of the Mann-Whitney U test, in combination with bootstrapping techniques (BMWU), as discussed in Section 2.3. Again, for each pixel we iteratively calculate 5000 bootstrap samples, where each sample has fixed size n. Moreover, for each pixel and sample, we randomly choose (with replacement) n predictions from the test set. Important to note is that the latter steps are simultaneously executed for each model of interest. The sample size n depends on whether a general ($n = 288$) or monthly ($n = 30$) BWMU analysis is conducted. Continuously, for each bootstrap sample we calculate the MSE, whereafter the obtained bootstrap distributions, for each model, are compared by means of the MWU test. However, taking in account the multiple testing problem, we control on the local false discovery rate (i.e. local fdr), resulting in adjusted p-values $\tilde{p}$ accordingly. The rationale for using the local fdr is easily explained by the fact that this technique is similar to the false discovery rate (FDR) procedure, providing less stringent control of Type I errors, compared to the familywise error rate (FWER) controlling procedures (e.g. Bonferroni correction), and hence increasing power. Moreover, the advantage of using the local fdr over the FDR procedure, is further given as more flexible interpretation of individual cases [8]. Finally, we construct a BMWU *significance map* $\mathbf{P}$, where each pixel/adjusted p-value $\tilde{p}_{y,x}$ is binarized as

$$\mathbf{P}[y,x] = \begin{cases} 0 & \text{if } \tilde{p}_{y,x} > 0.05 \\ 1 & \text{if } \tilde{p}_{y,x} \leq 0.05 \end{cases}$$

with y, x the height and width index, respectively. In Figure 3.11, significance maps are shown where extended $(\mathbf{Y}, \mathbf{X}_p, \mathbf{X}_t)$ and extended $(\mathbf{Y}, \mathbf{X}_p, \mathbf{X}_t^R)$ models are compared with the baseline $(\mathbf{Y})$ model. Recalling Section 3.3.2.1, we similarly conclude that the inclusion of raw temperature results in lower prediction performance, in contrast to the extended model with residual temperature included. Significantly better predictions are obtained in regions with low vegetation (e.g. Sahara desert/Africa), compared to baseline, whereas when looking at regions characterised with more vegetation (e.g. Amazon rainforest), no signifcantly better predictions are observed for the extended model with raw temperature. When it comes to the extended $(\mathbf{Y}, \mathbf{X}_p, \mathbf{X}_t^R)$ model, we observe significantly better predictions, compared to the baseline model, in most regions. Furthermore, in Figure 3.12 we compare the extended model with precipitation to the baseline model, for raw and residual vegetation data, respectively. In contrast to the raw vegetation models, we observe that the extended $(\mathbf{Y}^R, \mathbf{X}_p)$ model is only

significantly better than baseline for regions characterised by low vegetation. Hence, again, the issue with residual vegetation data, for the temporal convolutional network. Looking at Figure 3.13, illustrates that the extended $(\mathbf{Y}, \mathbf{X}_t^R)$ model is in general not signicantly better compared to the extended $(\mathbf{Y}, \mathbf{X}_p)$ model. We refer the reader to Appendix B.2.1, for additional and similar results. Finally, when looking at the monthly BMWU analysis results, in Section B.2.2, we observe that prediction performances are seasonal dependent. Indeed, when comparing the extended $(\mathbf{Y}, \mathbf{X}_p)$ model with baseline, it seems that for South America, significantly better prediction performances are observed for the extended model, from January until March (see Figure B.12), whereas from April until June (see Figure B.13), the opposite seems to be true.

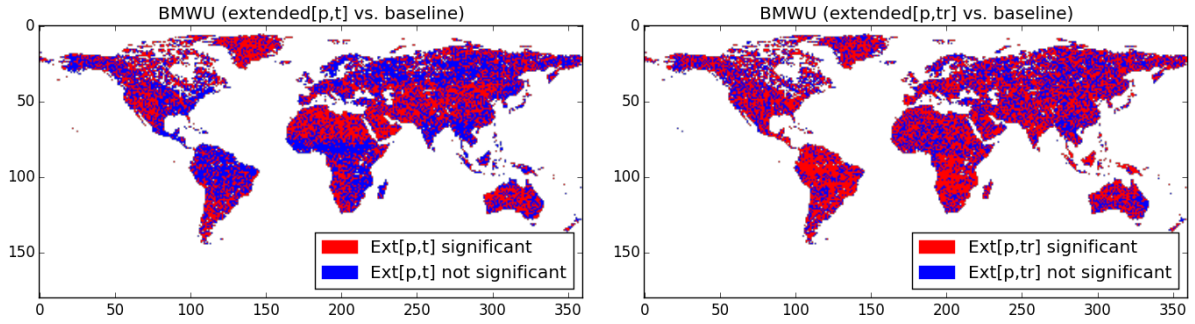


Figure 3.11: General BMWU significance maps, comparing extended $(\mathbf{Y}, \mathbf{X}_p, \mathbf{X}_t)$ and extended $(\mathbf{Y}, \mathbf{X}_p, \mathbf{X}_t^R)$ models with the baseline $(\mathbf{Y})$ model. Red pixels indicate significantly better MSE prediction performances for the extended models, whereas blue pixels indicate the opposite.

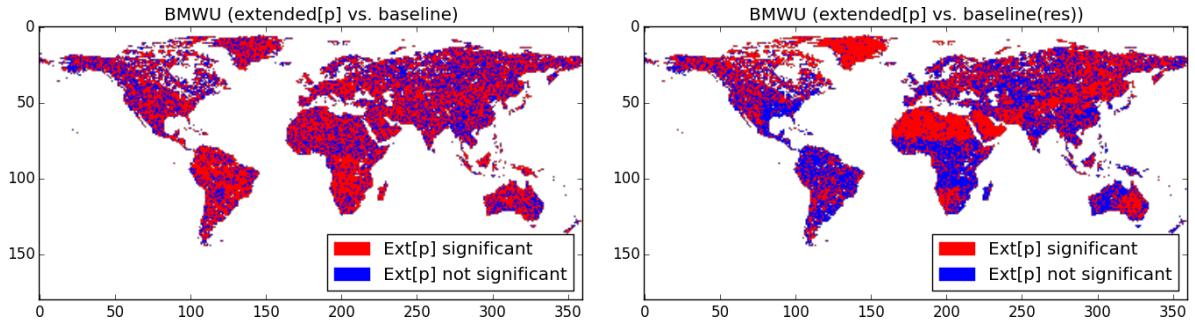


Figure 3.12: General BMWU significance maps, comparing baseline $(\mathbf{Y})$ and baseline $(\mathbf{Y}^R)$ models with extended $(\mathbf{Y}, \mathbf{X}_p)$ and extended $(\mathbf{Y}^R, \mathbf{X}_p)$ models, respectively. Red pixels indicate significantly better MSE prediction performances for the extended model, whereas blue pixels indicate the opposite.

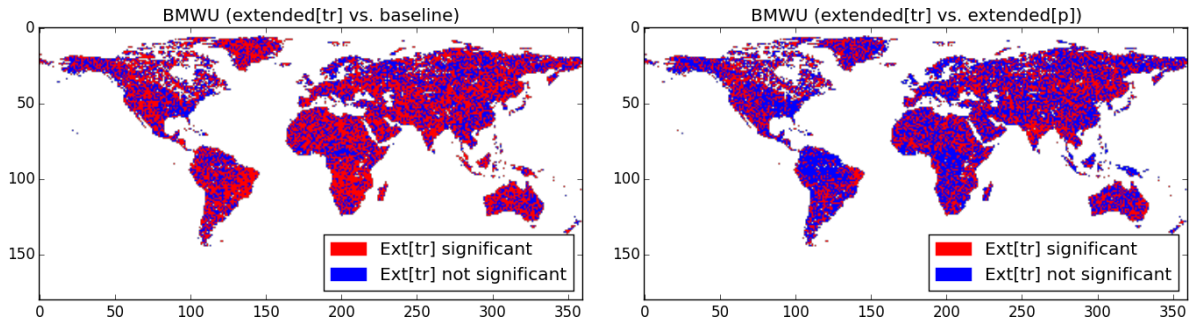


Figure 3.13: General BMWU significance maps, comparing baseline $(\mathbf{Y})$ and extended $(\mathbf{Y}, \mathbf{X}_p)$ models with the extended $(\mathbf{Y}, \mathbf{X}_t^R)$ model. Red pixels indicate significantly better MSE prediction performances for the extended $(\mathbf{Y}, \mathbf{X}_t^R)$ model, whereas blue pixels indicate the opposite.

4

Conclusion

In this thesis, the modeling of climate-vegetation dynamics, by means of machine learning techniques, in a non-linear Granger causality framework was studied. As a continuation of previous research by Decubber S., Papagiannopoulou C., et al. [6][23], we extended the non-linear Granger causality framework, with deep learning data-driven models, in order to enhance multitask learning, by means of exploiting information between neighbouring pixels, as well as global scale learning.

In order to model climate-vegetation dynamics, while taking in account the temporal coverage for the available data, a limited dataset was used, including precipitation (MSWEP), temperature (ERA) and vegetation (GLASS). Moreover, we used both raw and residual vegetation data, as target variable of interest, together with two climatic features/predictors (i.e. precipitation and temperature), and for which a Granger cause-effect relationship was assessed, between the latter features and target vegetation. Due to the immense succes of neural networks in many pattern recognition domains, we proposed different spatio-temporal neural networks, such as locally/convolutional recurrent networks, convolutional LSTM networks and more importantly temporal convolutional networks, for data-driven modeling. Moreover, in this thesis, we discussed the rationale for using the less complex non-recurrent temporal convolutional networks (see Section 3.2.3). Furthermore, these networks were trained, by means of a conservative sequential training-validation approach, while slightly diverging from a traditional Granger causality analysis. That is, in contrast to the Granger causality test, typically applied on individual time series, we worked with global input and output maps (i.e. images). Each input

sample was constructed by using past states of vegetation, as well as past states of different feature maps, whereas each output represents a future state map of vegetation. The rationale for the conservative training-validation approach was discussed, where we stated that standard approaches, such as K-fold cross-validation, are not feasible as each complex model would need to be retrained for each fold, as well as the fact that correlation would be present between training and validation folds.

Finally, depending on the included features (i.e. precipitation or temperature) and used targets (i.e. raw or residual vegetation), different baseline and extended models were tested and evaluated on a separate test set, by means of conducting out-of-sample MSE and bootstrapped Mann-Whitney U analysis. Bootstrapping techniques were used, in order to obtain different test sets, without the need for retraining each model. We then compared the MSE distributions of different baseline and extended models, by means of the Mann-Whitney U test, while taking in account the multiple testing problem, by controlling the local false discovery rate. Hence, the latter techniques allowed us to conduct in-depth evaluations for different baseline and extended models, within the spatio-temporal domain of the test set.

4.1 Summary

Optimal prediction performance was observed for the extended model with included precipitation feature. Therefore, a significant Granger cause-effect between precipitation and vegetation was concluded, in contrast with the raw temperature feature, characterised by strong cyclic and similar behaviour to vegetation. Moreover, compared to the baseline model, we did not observe signficantly better performance for the extended model with raw temperature as predictor. The opposite was observed when replacing raw temperature with residual temperature. Hence, information within the temperature residuals are more informative for the prediction of vegetation. Additionally, when comparing the average predictions with the corresponding global average, better performance was observed, in contrast to local average predictions. This was not entirely suprising, recalling the global scale learning property of the latter networks. Bootstrapped Mann-Whitney U analysis further indicated seasonal dependency in the prediction performances, for each extended model, and hence indicating that the different features may have seasonal-dependent influences on vegetation. Most importantly, it was shown that extended and baseline models for target (and feature) residual vegetation, yielded low out-of-sample R^2 performance, in contrast to models where raw vegetation was used. A possible explanation for this issue was given by the fact that residual vegetation maps are characterised by absence of visual structures/blobs. Moreover, as useful deep feature extraction, by means of consecutive convolutional-pooling layers, heavily depends on the presence of visible structures, we argued that these networks will most likely fail on residual vegetation data, and hence making subsequent G-causal inference useless. At the other hand, we emphasized that the non-stationary

behaviour, underlying the raw vegetation time series, can also obscur subsequent G-causal inference for different models with raw vegetation. However, when more data and computational resources are available, complex temporal/recurrent convolutional networks might be considered, in order to capture residual information, which resides within the raw vegetation data.

4.2 Future work

With the increasing amount of available data in climate science, or more specific climate change attribution research, the need for better/complex data-driven models is essential, in order to allow new opportunities for research and industry, as well as gathering novel insights. Due to its huge success, deep learning can be ideally used as an alternative to less flexible statistical models. However, one needs to be aware of the corresponding black box nature, together with limited statistically relevant inference. In this thesis, a compromising statistical-deep learning approach was proposed, by using deep learning techniques within a non-linear Granger causality framework. Taking in account the results, obtained throughout this thesis, we finally conclude with a brief summarization of perspectives, for further research:

- Use more samples, together with deep convolutional and recurrent layers (e.g. convolutional LSTM networks), in order to enhance the modeling of residual information within target and feature time series.
- When data is scarce, Bayesian neural networks might be preferred in order to avoid overfitting.
- Focus on alternative deep-statistical learning frameworks, by means of further research in statistical evaluation methods or deep learning methods.
- Combine statistical learning and deep learning models, in order to allow and improve relevant statistical inference.

References

[1] Lasagne: A lightweight library to build and train neural networks in Theano. http://lasagne.readthedocs.io/en/latest/, 2017. Accessed: 2017-06-18.

[2] Theano: A Python framework for fast computation of mathematical expressions. http://arxiv.org/abs/1605.02688, 2017. Accessed: 2017-06-18.

[3] Britannica. Machine learning – Artificial intelligence. http://www.britannica.com/EBchecked/topic/1116194/machine-learning, 2017.

[4] B. Christopher, M. *Pattern Recognition and Machine Learning*, volume 1. Springer, 1 edition, 2009. Chapter 5: Neural Networks.

[5] H. D. and W. T. Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology (London), 195, 215–243*, 1968.

[6] S. Decubber, C. Papagiannopoulou, W. Waegeman, and O. Thas. Spatiotemporal optimization of Granger causality methods for climate change attribution. 2017.

[7] I. Ebert-UPhoff. The potential of causal discovery methods in climate science. *NCAR CISL presentation, National Center for Atmospheric Research*, 2015.

[8] B. Efron. Local false discovery rates, 2005.

[9] J. Faghmous and V. Kumar. *A big data guide to understanding climate change: The case for theory-guided data science.* . Big data, 2(3), 2014.

[10] M. Fay and M. Proschan. Wilcoxonmannwhitney or t-test? on assumptions for hypothesis tests and multiple interpretations of decision rules. *Statistics Surveys. 4: 139*, 2010.

[11] H. G.F. On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory 14 (1): 55–63. doi:10.1109/TIT.1968.1054102.*, 1968.

[12] C. W. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, page 424438, 1969.

[13] A. Graves. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850, 2013.

[14] A. Gruslys, R. Munos, I. Danihelka, M. Lanctot, and A. Graves. Memory-efficient backpropagation through time. *CoRR*, abs/1606.03401, 2016.

[15] L. Helmut. *New introduction to multiple time series analysis (3 ed.)*. Berlin: Springer, 2005.

[16] S. Hochreiter and J. Urgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[17] James, G. and Witten, D. and Hastie, T. and Tibshirani, R. *An Introduction to Statistical Learning*. Springer, 2013.

[18] F. Jerome H. Data mining and statistics: What's the connection? *Computing Science and Statistics 29 (1): 3–9.*, 1998.

[19] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Effiicient backprop. In *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, pages 9–50, London, UK, UK, 1998. Springer-Verlag.

[20] A. C. Lozano, N. Abe, H. Li, A. Niculescu-Mizil, Y. Liu., C. Perlich, and J. Hosking. Spatial-temporal causal modeling for climate change attribution. *In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM*, page 587596, 2009.

[21] A. C. Lozano, N. Abe, Y. Liu, and S. Rosset. Grouped graphical granger modeling methods for temporal causal modeling. *15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM*, page 577586, 2009.

[22] H. Mann and D. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics 18 (1): 5060*, 1947.

[23] C. Papagiannopoulou, D. G. Miralles, N. E. Verhoest, W. A. Dorigo, and W. Waegeman. A non-linear granger causality framework to investigate climatevegetation dynamics. *Geosci. Model Dev. Discuss.*, 2016.

[24] S. Phil. Too big to ignore: The business case for big data. *p. 89. ISBN 978-1-118-63817-0*, 2013.

[25] SAS. Machine learning: What it is and why it matters. `www.sas.com`, 2017.

[26] Watson, D.J. *Comparative physiological studies on the growth of field crops: I. Variation in net assimilation rate and leaf area between species and varieties and within and between years. Annals of Botany*. Annals of Botany, 1947.

[27] P. Werbos. Backpropagation through time: what does it do and how to do it. In *Proceedings of IEEE*, volume 78, pages 1550–1560, 1990.

Appendices

A

Software and hardware specifications

When it comes to implementation, we used the following software:

- **Python 2.7.11 (NumPy, SciPy etc.)**
- **Theano 0.9.0.dev-RELEASE**
- **Lasagne 0.2.dev1**

Theano is a Python library and allows to define, optimize and evaluate mathematical expressions, involving multi-dimensional arrays, efficiently. The latter is accomplished by using a tight integration with NumPy, transparant use of a GPU, efficient symbolic differentiation, speed and stability optimizations and dynamic C code generation [2].
For code optimization and reduced computational complexity, we use Lasagne. This is a lightweight library to build and train various neural networks in Theano. Its main features consist of supporting many optimization methods, freely definable cost functions (no need to derive gradients due to Theano's symbolic differentiation) and of course a transparent support of CPUs and GPUs, due to Theano's expression compiler [1]. Moreover, we use a floating-point precision of 32 bits. Finally, we summarize the available hardware with corresponding specifications:

- **RAM: 8 GB 1600 MHz DDR3**
- **CPU: 2,9 GHz Intel Core i7**

B

Additional results

B.1 MSE analysis

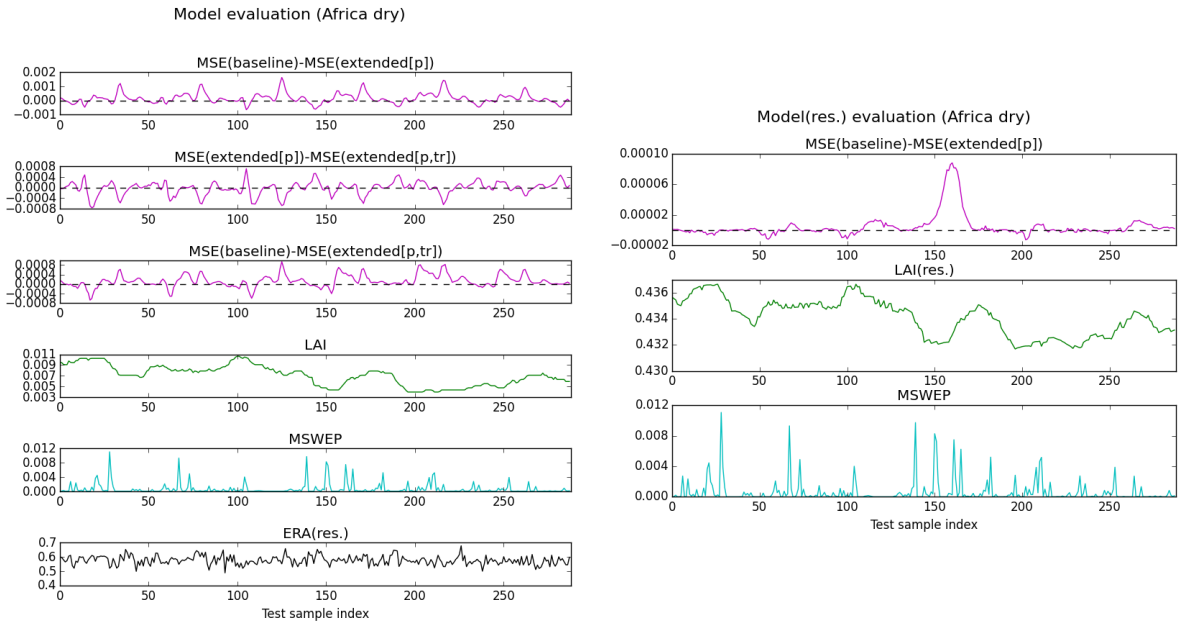


Figure B.1: MSE model evaluation for different models on local (i.e. Africa/Nigeria) scale, characterised by low vegetation. The left figure represents non-residual vegetation models, whereas the right represents residual vegetation.

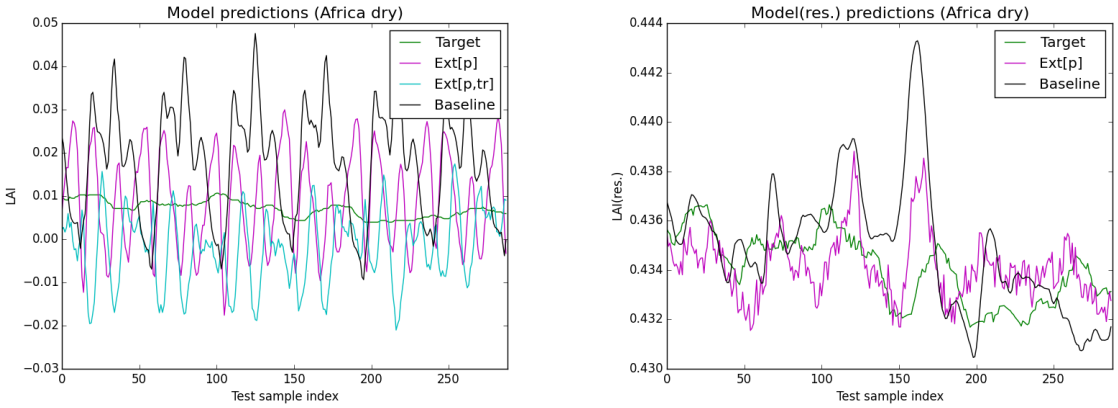


Figure B.2: Local (i.e. Africa/Nigeria) average predictions for different models, with corresponding targets, evaluated over the time domain of the test set. The left figure represents non-residual vegetation models, whereas the right represents residual vegetation.

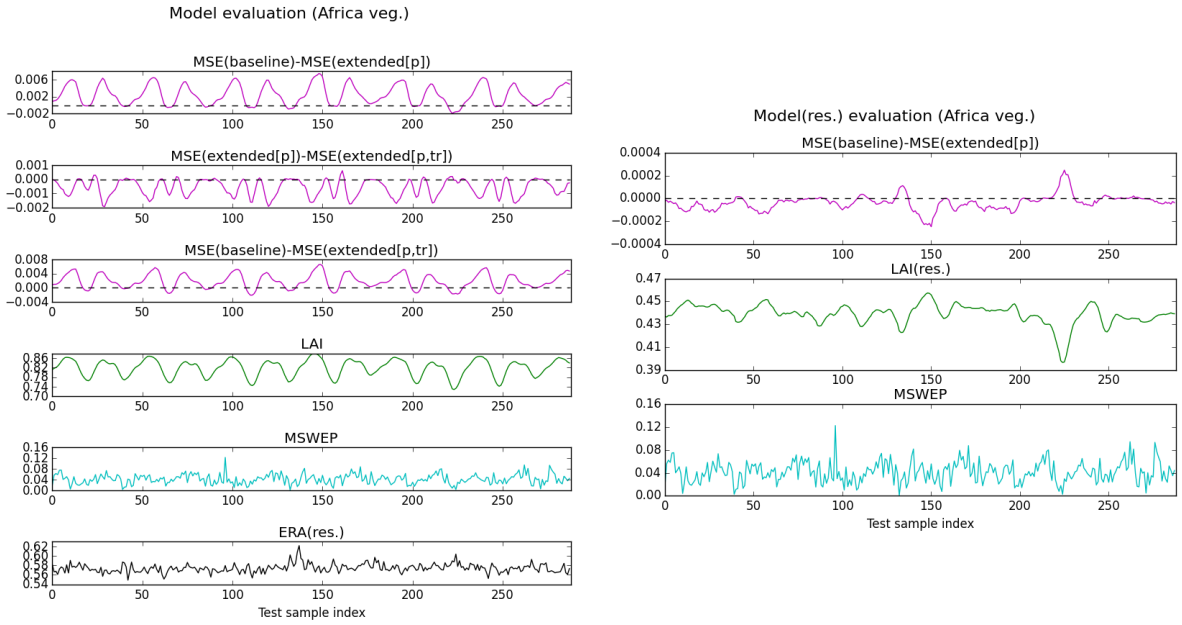


Figure B.3: MSE model evaluation for different models on local (i.e. Africa/Congo) scale, characterised by high vegetation. The left figure represents non-residual vegetation models, whereas the right represents residual vegetation.

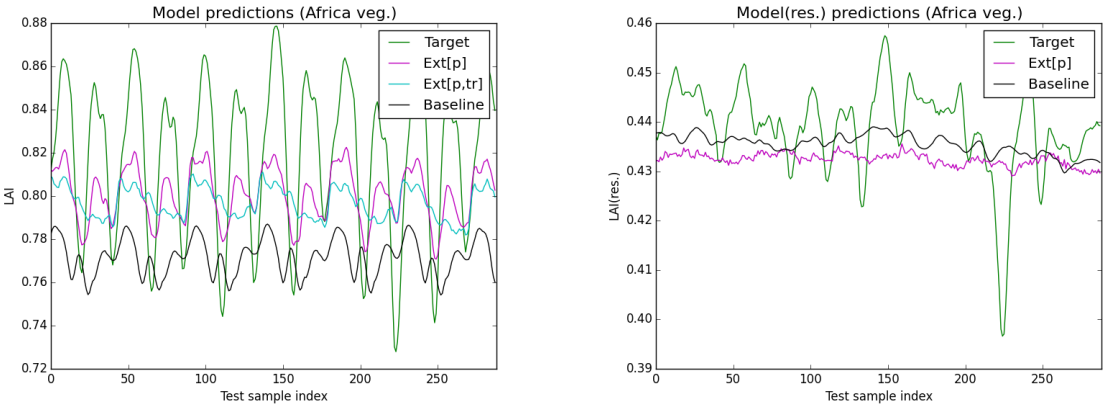


Figure B.4: Local (i.e. Africa/Congo) average predictions for different models, with corresponding targets, evaluated over the time domain of the test set. The left figure represents non-residual vegetation models, whereas the right represents residual vegetation.

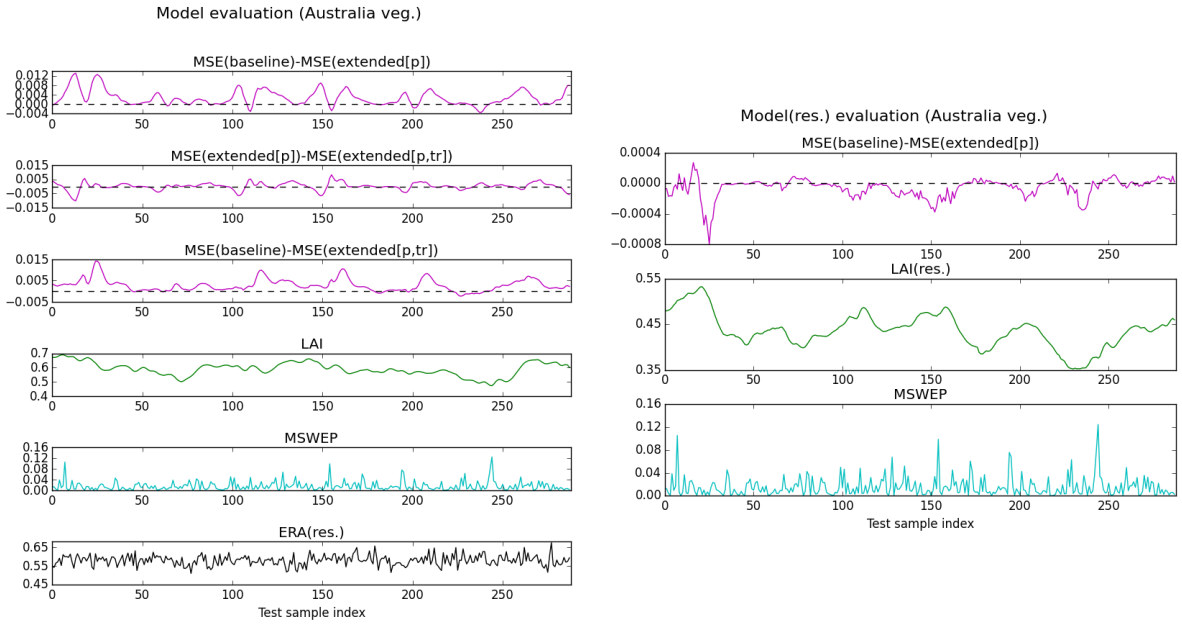


Figure B.5: MSE model evaluation for different models on local (i.e. Australia/Canberra) scale, characterised by high vegetation. The left figure represents non-residual vegetation models, whereas the right represents residual vegetation.

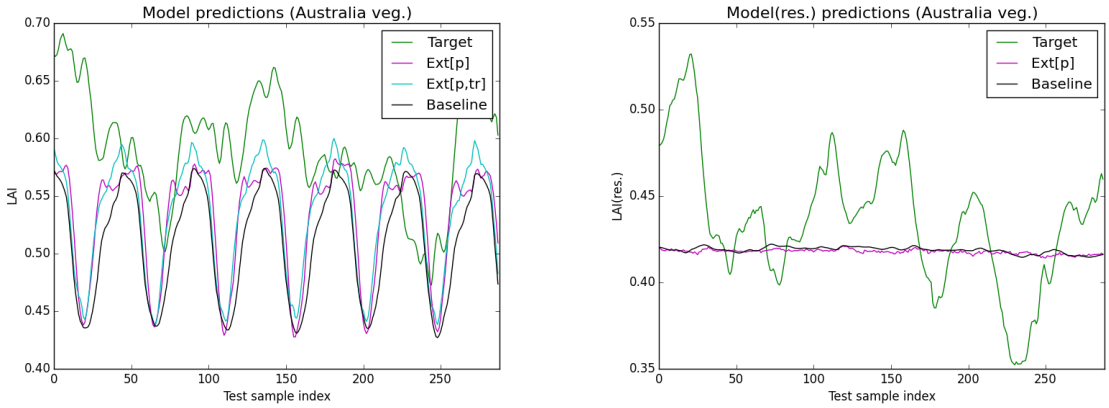


Figure B.6: Local (i.e. Australia/Canberra) average predictions for different models, with corresponding targets, evaluated over the time domain of the test set. The left figure represents non-residual vegetation models, whereas the right represents residual vegetation.

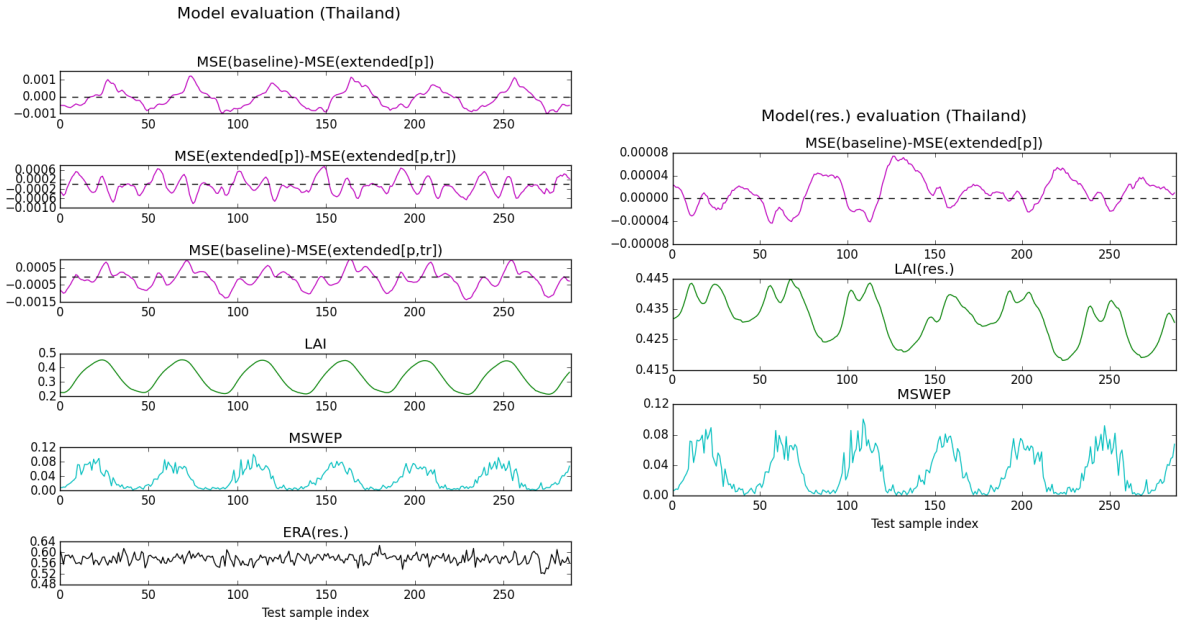


Figure B.7: MSE model evaluation for different models on local (i.e. Thailand) scale. The left figure represents non-residual vegetation models, whereas the right represents residual vegetation.

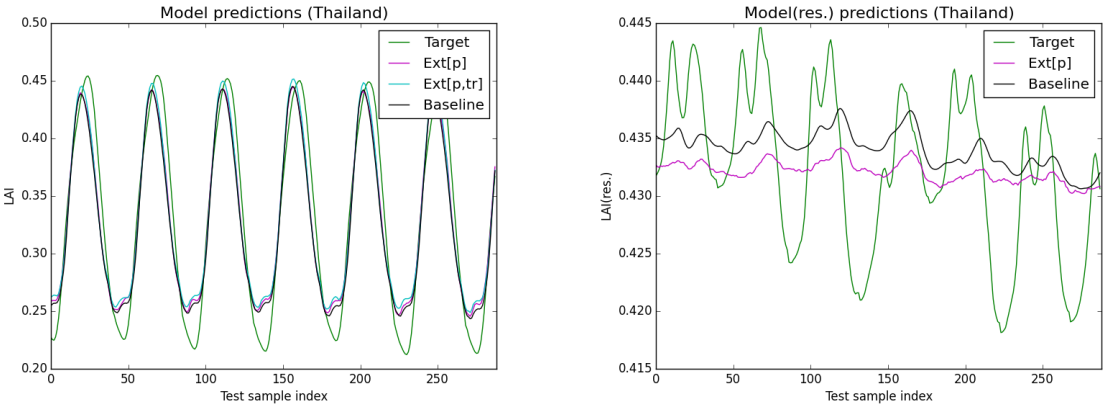


Figure B.8: Local (i.e. Thailand) average predictions for different models, with corresponding targets, evaluated over the time domain of the test set. The left figure represents non-residual vegetation models, whereas the right represents residual vegetation.

B.2 BMWU analysis

B.2.1 General

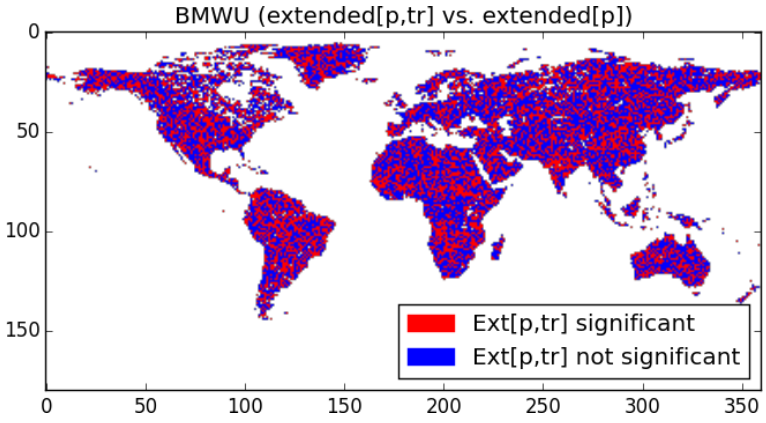


Figure B.9: General BMWU significance map, comparing the extended $(\mathbf{Y}, \mathbf{X}_p, \mathbf{X}_t^R)$ *with extended* $(\mathbf{Y}, \mathbf{X}_p)$ *model.*

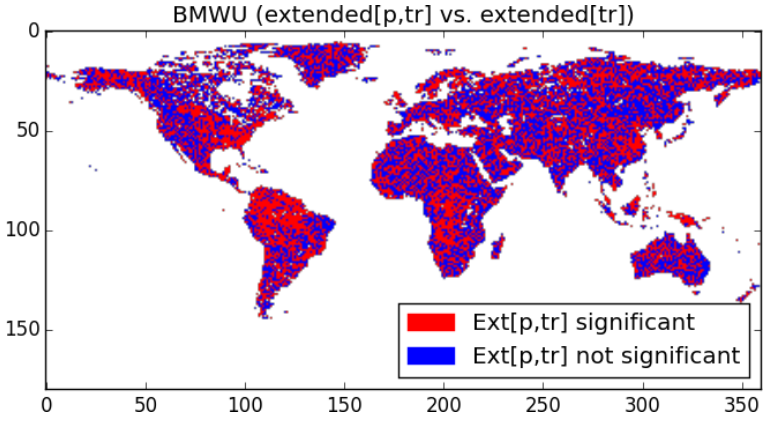


Figure B.10: General BMWU significance map, comparing the extended $(\mathbf{Y}, \mathbf{X}_p, \mathbf{X}_t^R)$ *with extended* $(\mathbf{Y}, \mathbf{X}_t^R)$ *model.*

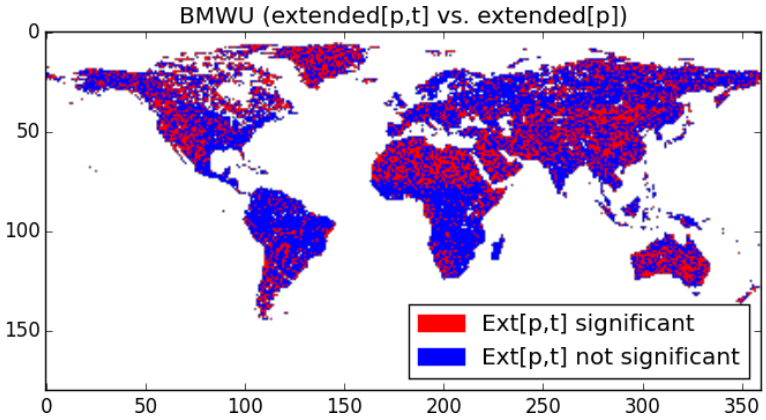


Figure B.11: General BMWU significance map, comparing the extended $(\mathbf{Y}, \mathbf{X}_p, \mathbf{X}_t)$ *model with extended* $(\mathbf{Y}, \mathbf{X}_p)$ *model.*

B.2.2 Monthly

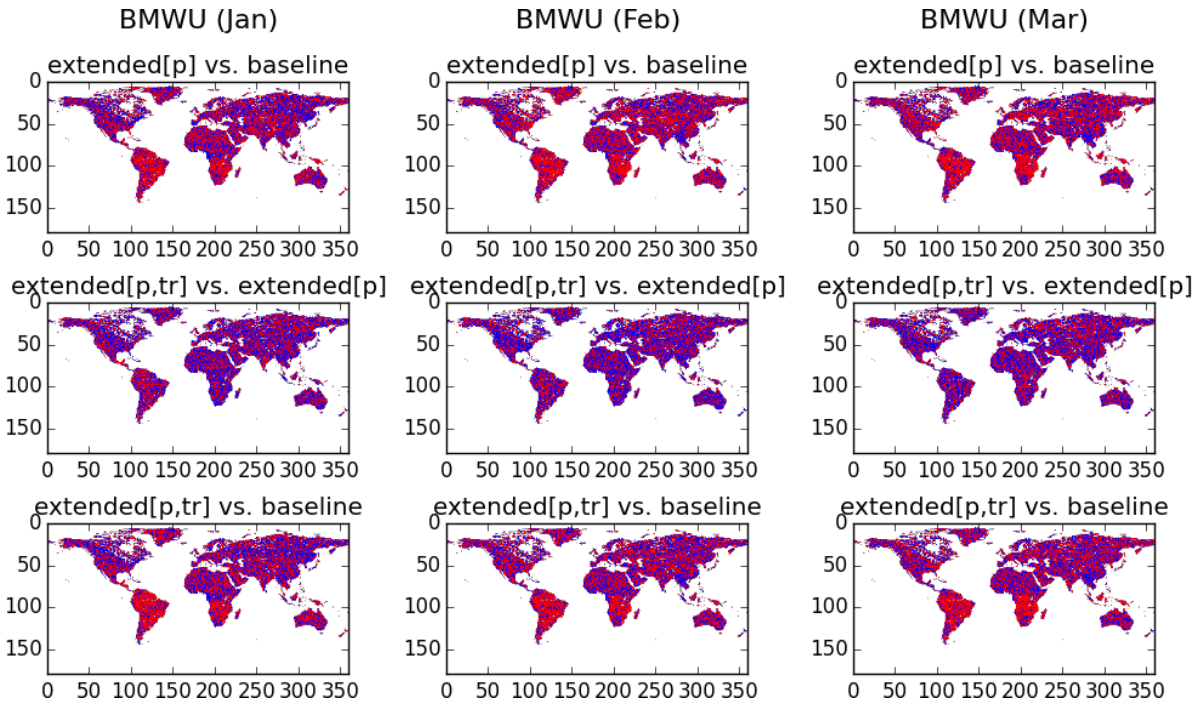


Figure B.12: Monthly BMWU significance maps, where red pixels indicate significance for the first model, obtained for January, February and March.

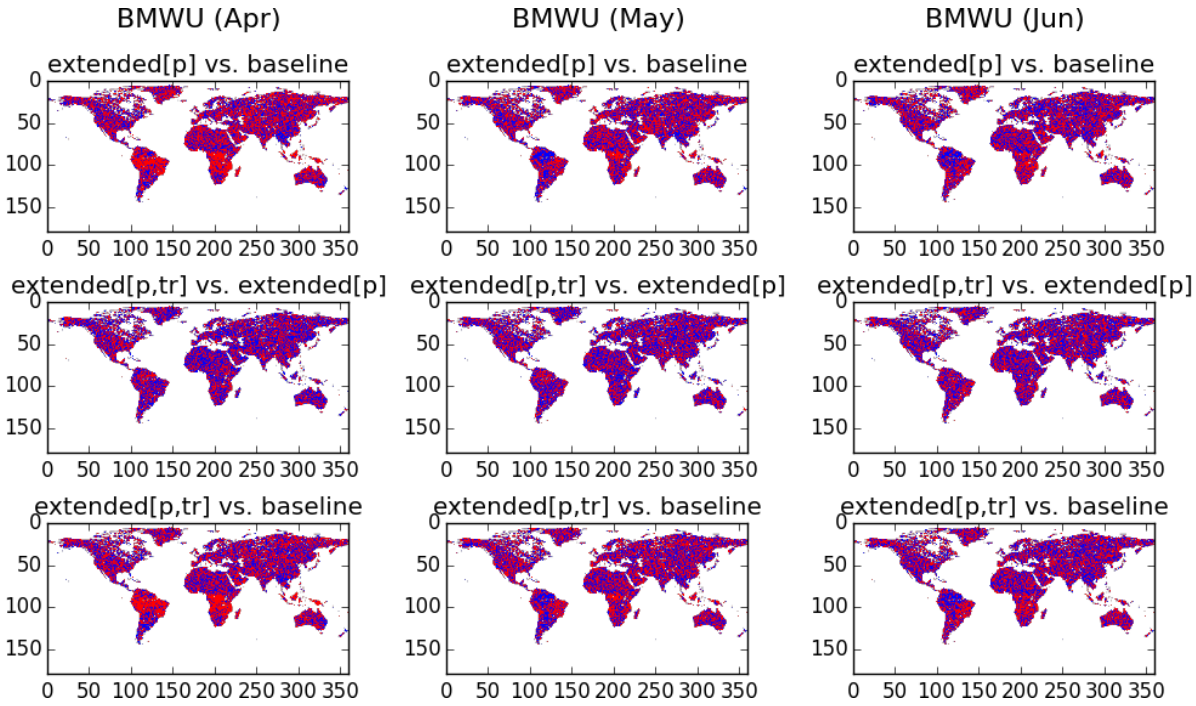


Figure B.13: Monthly BMWU significance maps, where red pixels indicate significance for the first model, obtained for April, May and June.

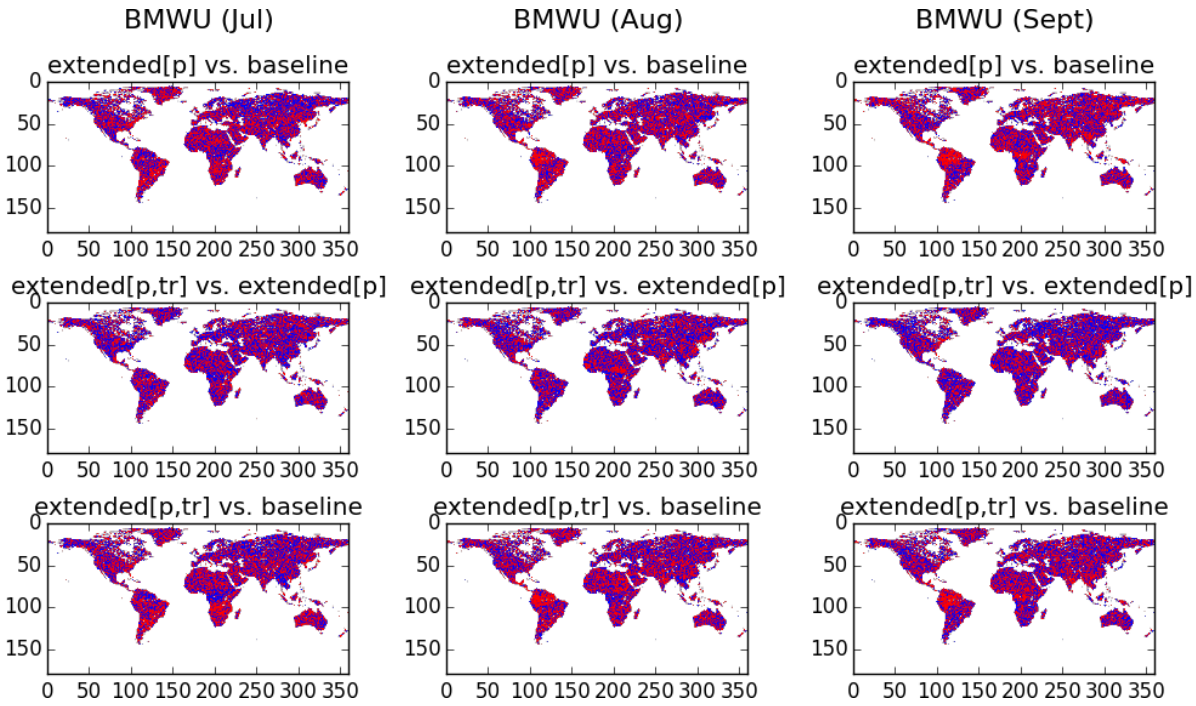


Figure B.14: Monthly BMWU significance maps, where red pixels indicate significance for the first model, obtained for July, August and September.

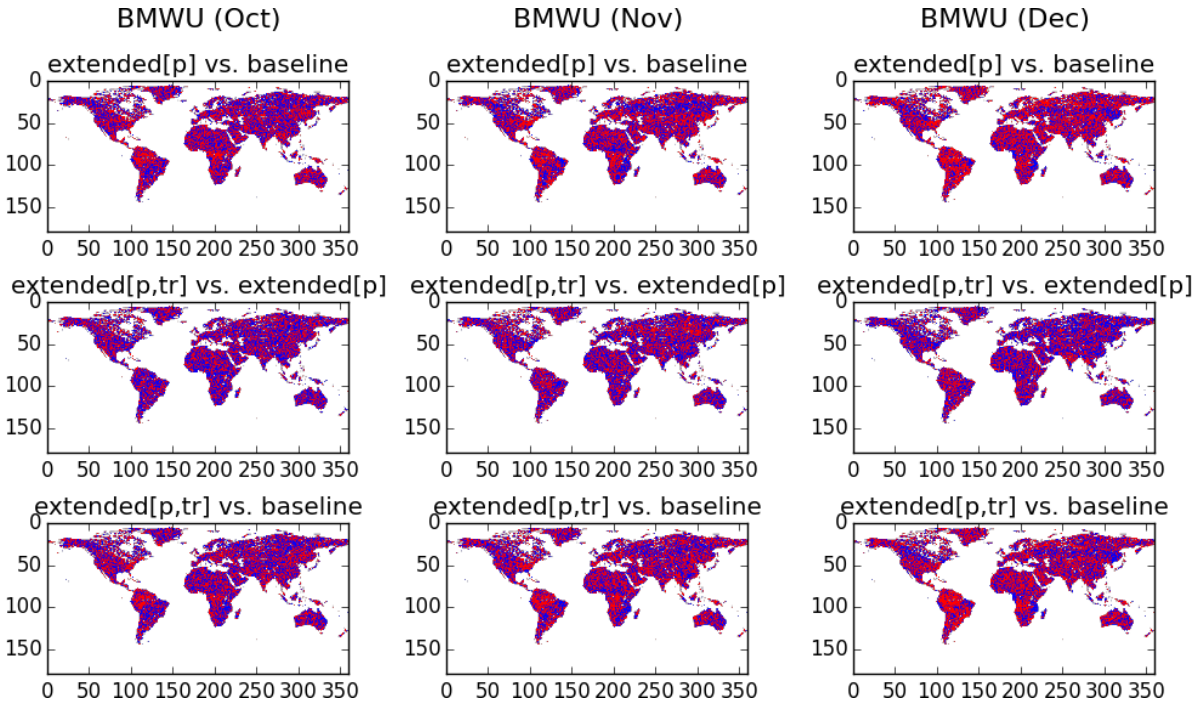


Figure B.15: Monthly BMWU significance maps, where red pixels indicate significance for the first model, obtained for October, November and December.