FACULTY OF
BIOSCIENCE ENGINEERING

UPDATING A CONCEPTUAL RAINFALL-RUNOFF MODEL BASED ON RADAR OBSERVATIONS AND MACHINE LEARNING

Olivier Bonte
Student ID: 01807260

Supervisors: Prof. dr. ir. Niko Verhoest, dr. ir. Hans Lievens
Tutor: dr. ir. Hans Lievens

A dissertation submitted to Ghent University in partial fulfilment of the requirements for the degree of master in Bioscience Engineering.
Academic year: 2022 - 2023

GHENT
UNIVERSITY

Gent, May 25, 2023

The author,	The promotors,
Olivier Bonte	Prof. dr. ir. Niko Verhoest, dr. ir. Hans Lievens

DANKWOORD

Merci aan allen!

CONTENTS

Dankwoord i

Contents v

List of acronyms vii

Nederlandse samenvatting xi

Summary xiii

1 Introduction 1
- 1.1 The role of machine learning in hydrology . . . 1
- 1.2 Incorporating observations in hydrological models: from data assimilation to hybrid modelling . . . 2
- 1.3 Objectives of this research . . . 3
- 1.4 An introductory note on open science and data . . . 4

2 Study area and data 5
- 2.1 Zwalm catchment . . . 5
- 2.2 Forcing data . . . 5
- 2.3 Flow data . . . 9
- 2.4 Satellite data . . . 11
  - 2.4.1 Land use data . . . 11
  - 2.4.2 Vegetation data: PROBA-V and Sentinel-3 . . . 11
  - 2.4.3 Radar data: Sentinel-1 . . . 13

3 SAR: from Sentinel 1 to features 15
- 3.1 Theoretical background on radar remote sensing . . . 15
  - 3.1.1 From SLAR to SAR . . . 16
  - 3.1.2 Key (environmental) factors affecting backscatter . . . 18
    - 3.1.2.1 Backscatter . . . 18
    - 3.1.2.2 Surface roughness . . . 19
    - 3.1.2.3 Electrical characteristics . . . 20
    - 3.1.2.4 Vegetation and urban environments . . . 20
    - 3.1.2.5 Influence of polarisation . . . 21
  - 3.1.3 Radar image characteristics . . . 21
- 3.2 Sentinel-1 . . . 21
  - 3.2.1 General information on satellite properties . . . 21
  - 3.2.2 Data products . . . 22
- 3.3 Pre-processing . . . 23
  - 3.3.1 GRD to σ^0 or γ^0_T processing . . . 23
  - 3.3.2 Spatial averaging of σ^0 or γ^0_T data . . . 24

4 The probability distributed model (PDM) 27
4.1 Introduction to rainfall-runoff modelling . . . 27
4.2 Description of the PDM . . . 28
4.3 Calibration . . . 30
4.3.1 Calibration algorithms . . . 31
4.3.1.1 Nelder-Mead algorithm . . . 32
4.3.1.2 Particle swarm optimisation . . . 33
4.3.2 Initial parameter set . . . 34
4.3.3 Results . . . 34
4.3.3.1 Nelder-Mead algorithm . . . 35
4.3.3.2 Particle swarm optimisation . . . 36
4.3.3.3 Comparison . . . 36

5 Machine learning methods for the inverse observation operator 41
5.1 The (inverse) observation operator for data assimilation . . . 41
5.2 Machine learning methods . . . 42
5.2.1 Linear methods . . . 43
5.2.1.1 Linear regression . . . 43
5.2.1.2 Ridge and lasso regression . . . 44
5.2.1.3 Support vector regression . . . 44
5.2.2 Nonlinear methods . . . 45
5.2.2.1 Nonlinear support vector regression . . . 45
5.2.2.2 Gaussian process regression . . . 46
5.2.3 Neural networks . . . 48
5.2.3.1 Multilayer perceptron . . . 48
5.2.3.2 Long short-term memory . . . 50
5.2.4 Software for the machine learning methods . . . 51
5.3 Feature engineering and exploration . . . 52
5.3.1 Features for time window methods . . . 52
5.3.2 Features and target correlations . . . 53
5.4 Experimental design: hyperparameters and model inputs . . . 54
5.4.1 Hyperparameter optimisation . . . 55
5.4.2 Input selection . . . 57
5.5 Results . . . 57
5.5.1 Linear methods . . . 57
5.5.1.1 Linear regression . . . 57
5.5.1.2 Lasso and ridge regression . . . 58
5.5.1.3 Linear support vector regression . . . 60
5.5.2 Nonlinear methods . . . 61
5.5.2.1 Nonlinear support vector regression . . . 61
5.5.2.2 Gaussian process regression . . . 61
5.5.3 Neural network methods . . . 62
5.5.3.1 Multilayer perceptron . . . 63
5.5.3.2 Long short-term memory . . . 63
5.5.4 Comparison and discussion of methods . . . 64

6 Data assimilation 67
6.1 Overview of methods for data assimilation . . . 67
6.2 Newtonian nudging . . . 68
6.3 Methodology . . . 69
6.4 Results . . . 70
6.4.1 Initial assimilations . . . 70
6.4.2 Comparison of data assimilation parameters . . . 72
6.4.3 Discussion . . . 74

7 Conclusions and future perspectives **75**

Bibliography **78**

Appendix A Supplementary material **93**

LIST OF ACRONYMS

E evaporation.

P precipitation.

PE potential evaporation.

h^{-1} inverse observation operator.

AI artificial intelligence.

API application programming interface.

BD72 Belgian Lambert 72.

CDF cumulative distribution function.

CI confidence interval.

CRS coordinate reference system.

CV cross-validation.

DA data assimilation.

dB decibel.

DEM digital elevation model.

DOY day-of-year.

EMR electromagnetic radiation.

ESA European Space Agency.

FDC flow-duration curve.

FHV percent bias in flow-duration curve high-segment volume.

GPR Gaussian process regression.

GRD ground range detected.

IW Interferometric Wide Swath.

L-BFGS limited-memory Broyden–Fletcher–Goldfarb–Shanno.

LAI leaf area index.

LaR lasso regression.

LR linear regression.

LSTM long short-term memory.

ML machine learning.

MLP multilayer perceptron.

mNSE modified Nash-Sutcliffe efficiency.

MSE mean squared error.

NM Nelder-Mead.

NN neural network.

NSE Nash-Sutcliffe efficiency.

OL open loop.

OLS ordinary least squares.

PDF probability density function.

PDM probability distributed model.

PSO particle swarm optimisation.

QQ quantile-quantile.

RBF radial basis function.

RNN recurrent neural network.

RR ridge regression.

RS remote sensing.

RTM radiative transfer model.

S-1 Sentinel-1.

SAR synthetic aperture radar.

SLAR side-looking airborne radar.

SLC single look complex.

SM soil moisture.

SSM surface soil moisture.

SVR support vector regression.

UTM Universal Transverse Mercator.

VMM Flemish Environment Agency.

WGS84 World Geodetic System 1984.

SAMENVATTING

Met een steeds toenemende impact van menselijke activiteiten op onze omgeving, zijn neerslag-afvoer modellen cruciaal om rivierdebiet te kunnen voorspellen op basis van hydrologische data zoals neerslag. Om deze modellen verder te verbeteren, is het interessant om observaties van hydrologische toestandsvariabelen (bv. bodemvocht) te gebruiken om het model te updaten via data-assimilatie. Wanneer fysisch-gebaseerde modellen gebruikt worden, zijn de toestandsvariabelen effectief observeerbaar. Als de observaties echter via teledetectie bekomen zijn, is er meestal alsnog een fysisch-gebaseerd *retrieval* algoritme nodig om de observaties, bv. radar terugverstrooiing, om te zetten naar een relevante toestandsvariabele.

Bij conceptuele modellen, zoals vaak gebruikt in operationele toepassingen, wordt assimilatie van observaties bemoeilijkt doordat de toestanden van het model niet eenvoudig gekoppeld kunnen worden met fysische eigenschappen op het terrein. Daarom is een observatie-operator nodig die de modeltoestand linkt met de observatie. Voor conceptuele modellen is deze relatie veelal empirisch. In deze thesis wordt onderzocht of a.h.v. *machine learning retrieval* algoritmes observaties bekomen via teledetectie rechtstreeks omgezet kunnen worden naar een conceptuele modeltoestand, specifiek de kritische capaciteit van het bodemvochtreservoir van het *probability distributed model* (PDM). Hiervoor worden Sentinel-1 observaties en de *leaf area index*, beiden uitgemiddeld per landsgebruik type van het beschouwde stroomgebied, als inputs gegeven aan de *machine learning* methoden om de kritische capaciteit in te schatten. Finaal worden deze data dan gebruikt om het PDM up te daten via data-assimilatie a.h.v. *Newtonian nudging*.

SUMMARY

With an ever increasing impact of human activity on our environment, rainfall-runoff models are of paramount importance to predict river discharge based on hydrological data such as precipitation. To further improve these models their performance, it is of interest to update them with observations of hydrological state variables such as soil moisture through data assimilation. When using physically-based models, these contain state variables that can effectively be observed. Nonetheless, if the observations are obtained via remote sensing, a physically-based retrieval algorithm is mostly still needed to convert the observation, for example radar backscatter, to a state of interest.

When using conceptual models, as is often the case for real-time forecasting, the problem becomes more complicated since the model states cannot be easily related to actual physical properties in the field. Therefore, an observation operator is needed which relates the model state to a hydrological observation. For conceptual models, this relation is often empirical. In this dissertation, it is investigated if remote sensing observations can be directly converted to a conceptual model state, i.e. the critical capacity of the soil moisture reservoir of the probability distributed model (PDM), by using machine learning methods as retrieval algorithms. For this purpose, Sentinel-1 observations and the leaf area index, both averaged out per land use type of the catchment, are utilised as inputs for the machine learning methods to estimate this critical capacity. Subsequently, these data are then used to update the PDM via data assimilation (i.e. Newtonian nudging).

1. INTRODUCTION

1.1 The role of machine learning in hydrology

Vrij grondig gewijzigd sinds ik dit laatste keer doorstuurde. Hoop dat dit nu een goede kadering is van de thesis, maar opmerkingen van jullie beiden zeker welkom.

With human activities having an ever growing impact on freshwater services, there is an increasing need for better predictions of the occurrence of freshwater in its different compartments, even when these are beyond the currently observed range (Wagener et al., 2010). For this purpose, there is a growing interest in the geoscientific community for the use of artificial intelligence (AI), which has known great success in other fields (such as natural language processing, computer vision, finance...) due to the increase in computational power and the availability of big data (Razavi, 2021). Also within the earth system sciences, the era of big data has arrived, with large data flows coming from both remote sensing (RS) and in-situ sensors (Reichstein et al., 2019). For this field of study, application of these new AI models often yields a better fit on the observed data when compared to the conventional process-based models, which are mostly based on the underlying physics. These data-driven models lack however in interpretability, explainability and ability to incorporate a priori knowledge on the system, which limit their widespread implementation (Razavi, 2021).

To tackle the aforementioned issues, there is the promising approach of combining process-based and data-driven models in hybrid modelling. In this way, the physical consistency of the former can be combined with the ability of the latter to extract useful insights from data (Reichstein et al., 2019). Note that this approach coincides with combining the two opposing approaches seen in rainfall-runoff modelling, the subject of this dissertation, where the goal is to predict river discharge based on available hydrological data such as precipitation. According to the first 'top-down' point of view, rainfall-runoff models should be seen as tools that allow extrapolation (and transformation) in space and time of the available measurements. For this purpose, the data-driven machine learning (ML) methods are well suited. The second 'bottom-up' approach, related to the process-based modelling point of view, uses within these models the available physical knowledge on the system to a maximum extent to give the best predictions outside of the observed range. Unfortunately, observations are prone to error while physical descriptions of hydrological processes will most likely never be fully non-empirical, which stresses the weaknesses of both methods when used separately (Beven, 2012a). To further

improve our ability to make informed decision in hydrological problems, which can be seen as the ultimate aim of using rainfall-runoff models (Beven, 2012b), the above mentioned hybrid method is therefore of interest.

To better frame this new hybrid modelling paradigm, it is of interest to first examine the current state of the art in purely data-driven rainfall-runoff modelling. Most notable here is the pioneering work of Kratzert et al. (2018), who were one of the first to use a special type of neural network (NN), the long short-term memory (LSTM) network (cf. Chapter 5), for discharge prediction. Although this initial work mainly focused on training one model per catchment, Kratzert et al. (2019b) prove that training one LSTM model on 531 catchments with their static catchment attributes as additional inputs, results in a regional rainfall-runoff model outperforming classical rainfall-runoff models trained on a single catchment.

With its black box nature, the LSTM suffers from the aforementioned lack of explainability and interpretability. Therefore attempts at interpreting the internals of the LSTM (Kratzert et al., 2019a) and making the networks mare physically plausible by including mass conservation (Hoedt et al., 2021) have been undertaken. It can be argued however that instead of explaining black box models, one should try to make interpretable models in the first place (Rudin, 2019). For this purpose, hybrid models show their potential. Recent works of interest here are the embedding of a simple rainfall-runoff model in a NN architecture by Jiang et al. (2020) and the combination of ordinary differential equations with NNs by Höge et al. (2022).

1.2 Incorporating observations in hydrological models: from data assimilation to hybrid modelling

The goal of optimally combining observational data with simulations by process-based models is not new however, as this is the central goal of data assimilation (DA) (Evensen et al., 2022). Although the mathematical discipline originates from the fields of oceanography and meteorology (Bouttier and Courtier, 2002), it is not limited to these applications, as it is also used in (for example) seismology, petroleum reservoir modelling and hydrology (Evensen et al., 2022). In hydrology, a large variety of observational data has already been assimilated in models for hydrological predictions, both measured in-situ (e.g. river discharge, soil moisture (SM) and snowpack measurements) as obtained via RS (Liu et al., 2012). Especially SM observations are of interest in rainfall-runoff modelling, as more accurate predictions of SM can lead to better modelling of other hydrological processes such as runoff, evaporation and groundwater recharge (Alvarez-Garreton et al., 2014).

When focusing on DA for the estimation of state variables of these process-based models, a key challenge is that the model states are often not directly observable.

Certainly in RS, these observations are often only indirectly related to the state variables. To be able to compare and combine observations and model states, either the latter need to be translated to the observation space with a (forward) observation operator, or the observations mapped to states with a retrieval algorithm/inverse observation operator. An example of an observation operator is a radiative transfer model (RTM), which translates geophysical states (e.g. SM) to radiances, which can then be compared with radiances measured via RS (Reichle, 2008).

When physically-based models are used, their state variables (e.g. temperature and pressure for meteorology) can be used in physically based observation operators such as the RTM (Kalnay, 2002). However in rainfall-runoff modelling, lumped conceptual models (cf. Chapter 4 for details) are often used for e.g. operational forecasting. These model structures aggregate the hydrological processes in space and time to a number of fluxes and storage elements (e.g. SM reservoir, groundwater compartment...), consequently leading to model states and parameters that can not directly be linked to (field) measurements (Solomatine and Wagener, 2011).

If physical state variables can be derived from the conceptual model structure, physically-based observation operators are still applicable directly (see for example Hostache et al. (2020) for the use of an RTM), but otherwise empirical relations have to be used. In rainfall-runoff modelling, simple relationships such as linear regression have already been used for SM assimilation (Aubert et al., 2003; Alvarez-Garreton et al., 2014). Note however that for the case of measurements via RS, a retrieval algorithm is still needed to obtain the SM estimates in the first place (see for example Owe et al. (2008)).

Instead of first retrieving SM from RS data and then applying an empirical relationship as (inverse) observation operator for the conceptual rainfall-runoff model, it can also be attempted to combine these two steps into one. For this purpose, more complex ML algorithms could serve as (inverse) observation operator. As illustrated by Siyang (2019), Rains et al. (2022) and Liang et al. (2023), a similar approach has already been applied to replace physically based or semi-empirical observation operators in the scientific fields of sea ice modelling, SM prediction and numerical weather prediction. Although stronger forms of hybrid modelling can be explored by linking ML and process-based/conceptual models directly for joint estimation of the outputs of interest (cf. supra), using ML with DA on process-based/conceptual models can be seen as a first step towards this hybrid paradigm.

1.3 Objectives of this research

In this master's thesis, the goal is to use ML as the inverse observation operator to estimate a state variable related to SM in a conceptual rainfall-runoff model called the probability distributed model (PDM). For this estimation, RS data obtained from

the Sentinel-1 (S-1) satellites will be used as an input for the data-driven model. Combined with data on vegetation and landuse, the goal is for the ML method to isolate the effects in the S-1 imagery related to soil moisture differences to get an optimal estimate of the state variable, which is called the critical storage capacity. Once this empirical relationship has been established, the data will be used to update/improve the model via a classic type of DA. It is hypothesised that this approach will improve river discharge predictions in the catchment.

This dissertation will start in Chapter 2 with a description of the study area, the Zwalm catchment, where there will be a focus on the different datasets used. As some insight in the radar data obtained from S-1 is of importance, an introduction to synthetic aperture radar (SAR) and the pre-possessing of this imagery will be given in Chapter 3. With all the data covered, Chapter 4 will focus on the the calibration of the PDM to optimally predict the catchment discharge without using the auxiliary RS data. The critical storage capacity from the PDM will then be used to train several ML methods in Chapter 5 for the retrieval of an "observed" critical storage capacity. Finally, Chapter 6 will cover the assimilation of the ML-retrieved critical storage capacity in the PDM and assess the potential merits of this approach.

1.4 An introductory note on open science and data

It is common knowledge that reproducibility of research is a key bottleneck in science, and therefore also in hydrology. By Stagge et al. (2019) it is estimated with 95 % confidence that only 0.6 to 6.8 % of the articles published in 2017 in six major hydrology and water resources journals are reproducible. This indicates that many of the modern tools available for conducting reproducible science are underutilised in the hydrological community.

In this dissertation, it will be attempted to provide open access to both code and data. All code used can be found in the following GitHub repository: `https://github.com/olivierbonte/master_thesis`. Furthermore Python (Van Rossum and Drake, 2009) will be used as programming language, as it is fully open-source and hence allows reproducibility without propriety software licenses. The repository includes a description of all software dependencies. The most important Python packages used that are not explicitly mentioned in one of the subsequent chapters, are: `SciPy` (Virtanen et al., 2020), `NumPy` (Harris et al., 2020), `Matplotlib` (Hunter, 2007), `GeoPandas` (Jordahl et al., 2022) and `Xarray` (Hoyer and Hamman, 2017). A full description on how to execute the code on a virtual machine via Binder (Project Jupyter et al., 2018) or on a local computer can be found in the aforementioned repository together with links to raw and generated data on Zenodo. Lastly, it must be mentioned that ChatGPT was used in this dissertation for the generation of minimal pieces of code (e.g. for visualisation).

2. STUDY AREA AND DATA

Dit stuk is door Niko al nagelezen en kleine veranderingen uit de opmerkingen zijn doorgevoerd. Minder noodzakelijk denk ik dat één van jullie 2 dit opnieuw naleest.

For this dissertation, one catchment is studied to serve as a proof of concept for the method proposed in Section 1.3. A brief description of this catchment is given in Section 2.1. Subsequently, the forcing data needed to simulate with the conceptual rainfall-runoff model, described in Chapter 4, is given in Section 2.2. Finally, the satellite data-products, which will serve as input for the ML methods of Chapter 5, are described in Section 2.4.

2.1 Zwalm catchment

The catchment of the river Zwalm is located in the province East-Flanders in Belgium and has an area of 115.21 km^2. According to Cabus (2008), Flemish catchments can be categorised in at least three groups. The Zwalm belongs to the group characterised by high peak flow (HQ) with high flow variation (HCV) called HQHCV (Cabus, 2008). HQHCV catchments have moderate slopes, as can be seen on the digital elevation model (DEM) in Figure 2.1. The displayed DEM is a clipped and masked version of the Copernicus Global 30 meter DEM dataset (Copernicus, 2022). In Figure 2.1, the Zwalm and its tributaries as found in the *Vlaamse Hydrografische Atlas* are displayed (Vlaamse Milieumaatschappij, 2022). To be in a common coordinate reference system (CRS) for visualisation, the DEM is reprojected from its original World Geodetic System 1984 (WGS84) CRS to Belgian Lambert 72 (BD72) by bilinear resampling. Note that unless otherwise specified, BD72 is used as CRS in what follows.

Besides being a prime example of a Flemish HQHCV catchment, the Zwalm is also chosen because of its frequent earlier use within the H-CEL research unit for testing of novel hydrological applications (cf. for example Pauwels et al. (2002) and Vernieuwe et al. (2003))

2.2 Forcing data

Common,important forcing data for rainfall-runoff models are precipitation (P), evaporation (E) and net radiation (Kratzert et al., 2023). For the PDM used in this dis-

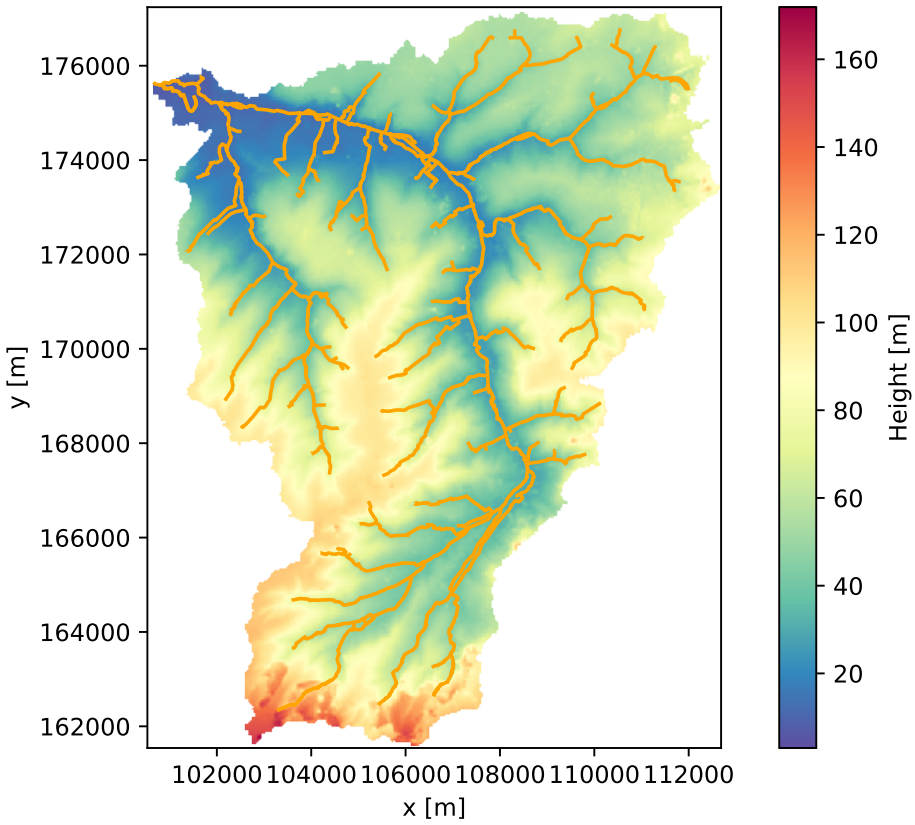

Figure 2.1: DEM of the Zwalm catchment (Copernicus, 2022). The river Zwalm and its tributaries (Vlaamse Milieumaatschappij, 2022) are displayed in orange.

sertation, only the first two are considered. In Flanders, both *P* and potential evaporation (*PE*) data can be obtained through `https://www.waterinfo.be/`. For this research, a Python application programming interface (API) called `pywaterinfo`, created by Van Hoey et al. (2021), is used to access the data on this site. The *PE* is calculated for meteorological stations based on the Penman-Monteith equation (cf. Monteith (1965)). Note that despite the term potential evapotranspiration being used on the waterinfo site, it is replaced by potential evaporation in what follows since the term evaporation is better suited to describe the full latent heat flux (soil evaporation, transpiration i.e. evaporation inside leaves and interception) (Miralles et al., 2020).

Both *P* and *PE* data are consulted between 01/01/2012 00:00 and 05/11/2022 23:00. *P* data are already in the desired time-resolution of one hour. *E*-data on the other hand have a 15 min resolution and consequently are resampled to an hourly basis by summation.

As precipitation varies over the catchment, one rainfall gauge cannot be representative of this entire area. Therefore, rainfall data are interpolated between stations. Although many interpolation options exist, the choice is made to use Thiessen polygons, as this method is also used by Flanders Hydraulic Research for modelling with the PDM (Maroy et al., 2021). This is a nearest neighbour method, assigning each

point in the catchment to its nearest gauge and consequently creating polygons encompassing the area assigned to each gauge. The rainfall from the different gauges are now weighted according to the relative area their polygon takes up within the catchment. With A_i and P_i respectively the area of the polygon and rainfall for gauge i, the average rainfall is calculated as:

$$\bar{P} = \frac{\sum_i A_i P_i}{A} \tag{2.1}$$

with A the area of the catchment (Thiessen, 1911).

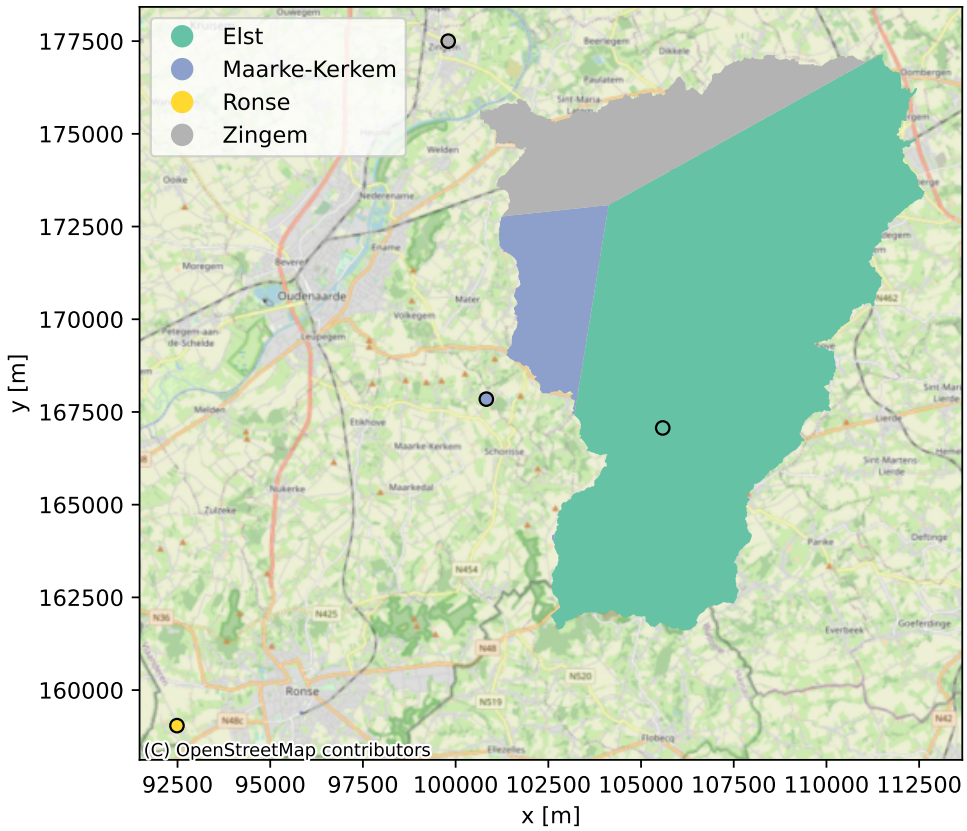

Figure 2.2: Rainfall gauges used for interpolation. Within the Zwalm catchment, Thiessen polygons are displayed for full data availability.

For P data, four gauges in the vicinity of the Zwalm are chosen (see Figure 2.2): Elst, Maarke-Kerkem, Ronse and Zingem. When data are available for all four stations, data from the station in Ronse is not used, as can be seen by the Thiessen polygons displayed in Figure 2.2. It is still useful to include this fourth station however, as in this way there is at least one gauge providing P data at every time step over the entire considered period. At each time step, it is checked how many of the four stations provide data and only for these ones Thiessen polygons are constructed and used in Equation 2.1. Consequently, different polygons than the ones displayed in Figure 2.2 are used at time steps with less data-availability. The time series of rainfall data from the gauges before and after interpolation are given in Figure 2.3.

It is apparent that by interpolation, the most extreme peaks in rainfall intensity (up to 30 mm/h before interpolation) are flattened.

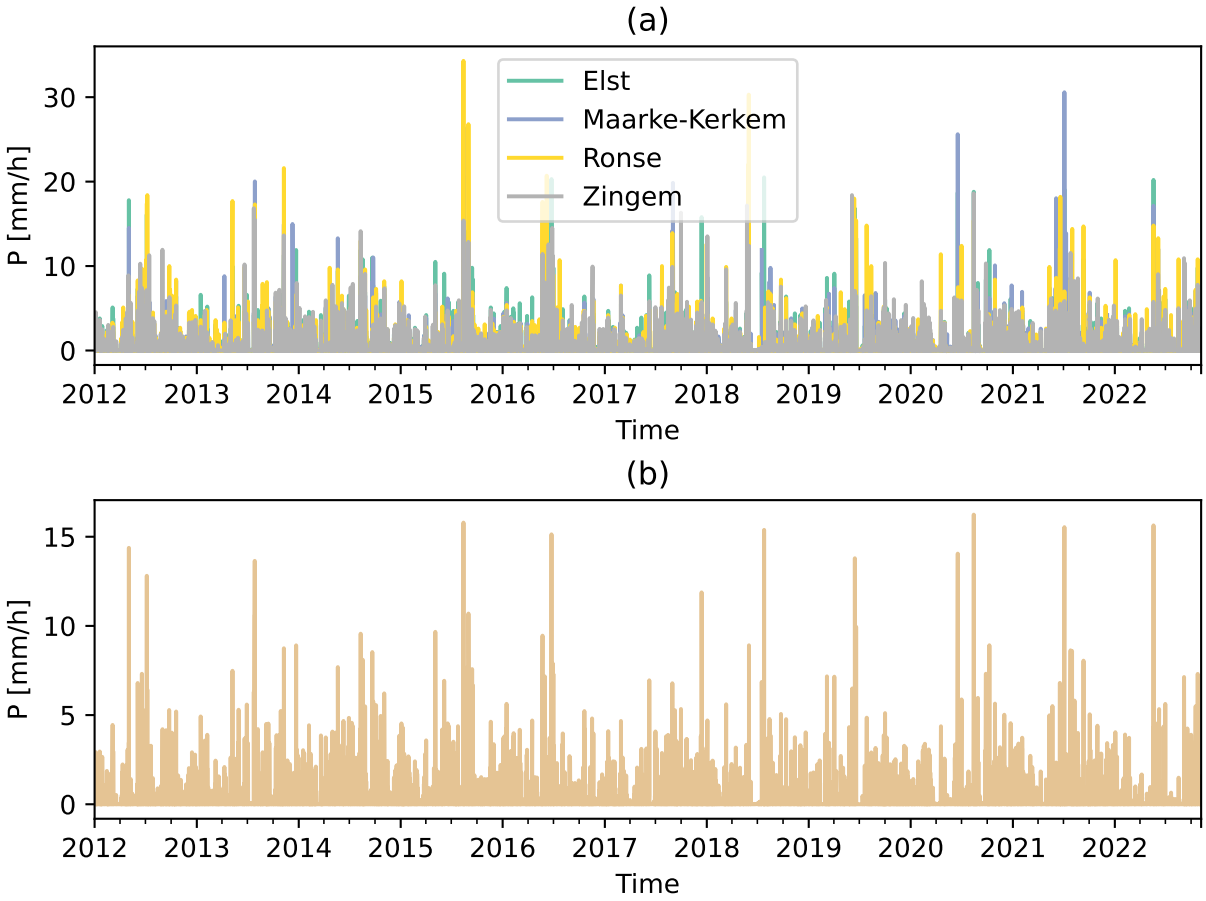

Figure 2.3: (a) Rainfall intensity from all four gauges. (b) Rainfall intensity after interpolation.

For *PE* data, an analogous approach is taken. Here, 3 meteorological stations in reasonable proximity of the catchment are chosen (see Figure 2.4): Boekhoute, Liedekereke and Waregem. When data are available for all stations, only two are used, whose polygons are displayed in Figure 2.4. As for rainfall, the seemingly superfluous third station is included to increase data availability in case of missing data at a gauge. Contrary to the explanation above however, there are time steps when none of the included stations provide *PE* data. To fill in these missing values, following procedure is followed:

1. Calculate a daily *PE* profile for all days of the year by taking the average of the *PE* values at each hour of every day over all years and excluding missing values.
2. Fill in the missing hours of that day with the corresponding values from its daily *PE* profile.
3. Apply a correction factor to the filled in values by multiplying them with the mean *PE* value over the known hours of that day divided by the mean value

over the same hours on the daily profile. This can be seen as a rescaling accounting for a deviation from the average over the years.

The time series of *PE* data before and after interpolation are given in Figure 2.5.

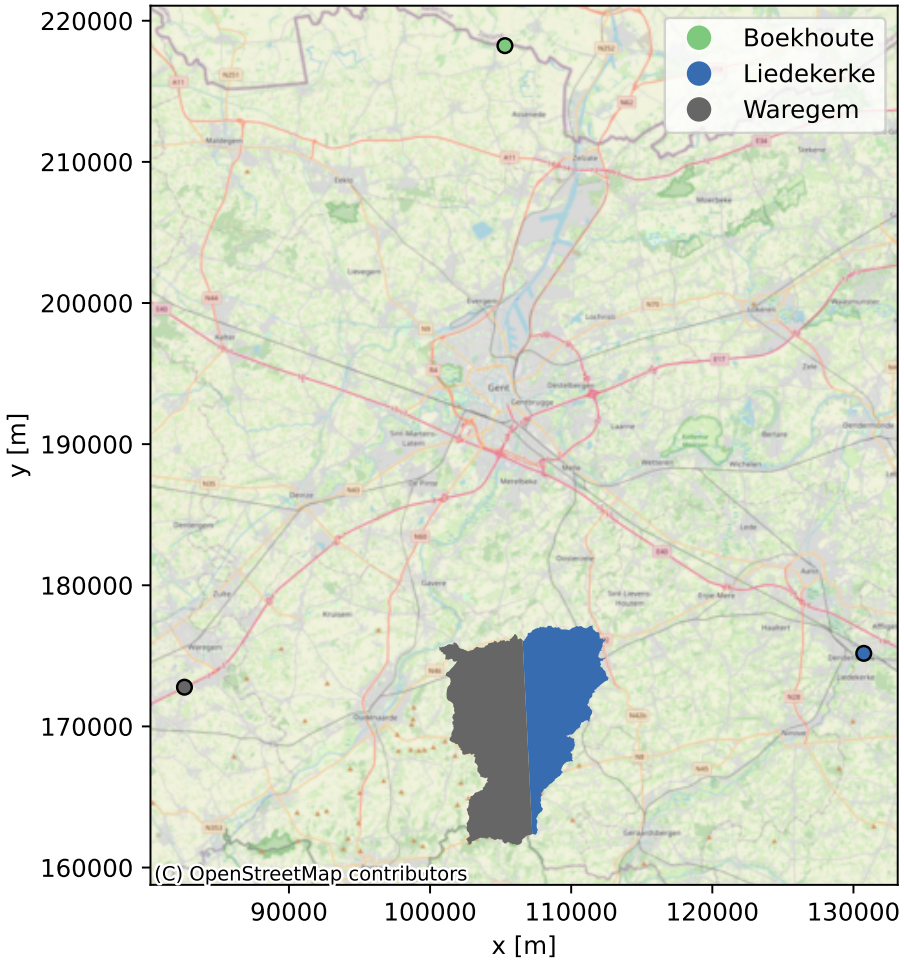

Figure 2.4: Meteorological stations used for *PE*. Thiessen polygons within the catchment for full data availability are displayed.

2.3 Flow data

As mentioned in Section 1.3, the eventual goal of this dissertation is to improve the predictions of river discharge by the PDM. Therefore, observed river flow is a crucial variable for evaluating the model performance. Through `pywaterinfo`, daily discharge data of the Nederzwalm/Zwalmbeek station, located near the mouth of the Zwalm, is retrieved for the same period as the forcing data. The data are visualised in Figure 2.6, where discharge [m^3/s] is presented as Q.

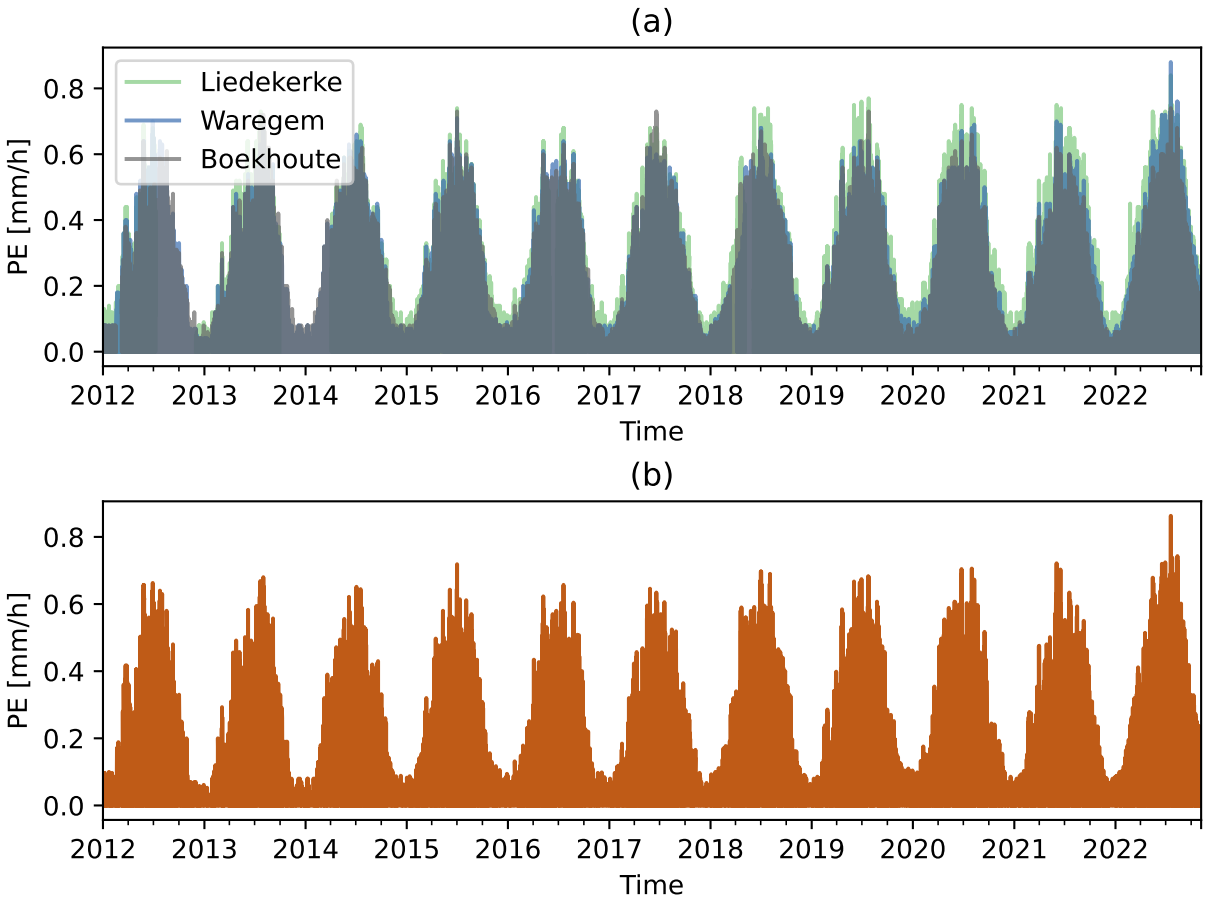

Figure 2.5: (a) *PE* from the three distinct meteorological stations. (b) *PE* after interpolation.

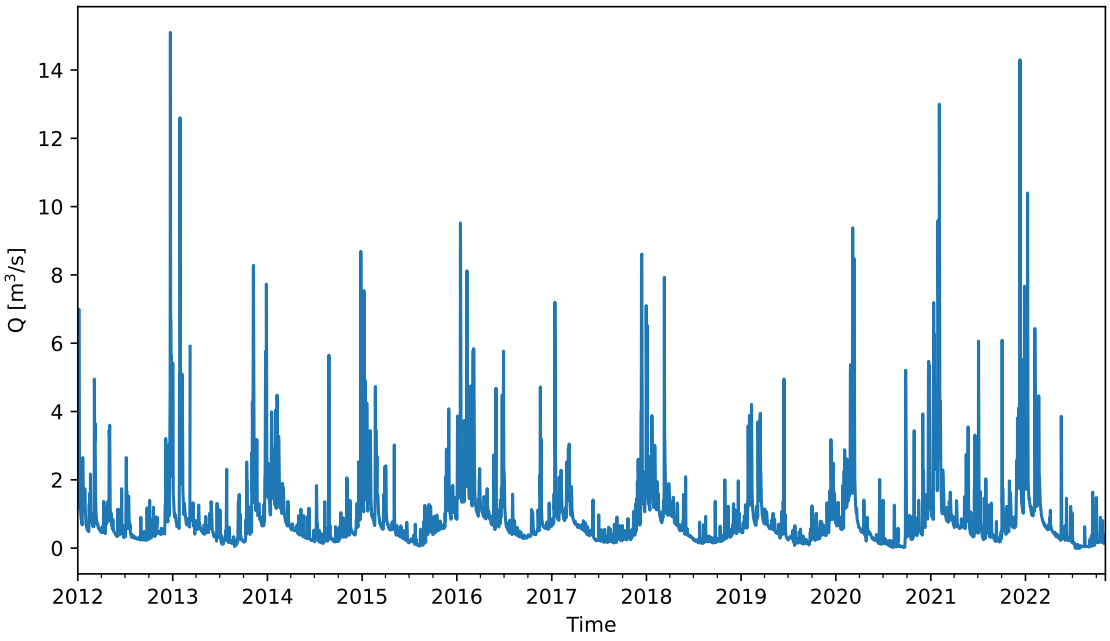

Figure 2.6: Daily observed discharge at the Nederzwalm/Zwalmbeek gauge.

2.4 Satellite data

With the open science commitment from Section 1.4 in mind, it is attempted to maximally use open-source API's for data retrieval. For this purpose, the openEO API, which was first described in literature by Schramm et al. (2021), is used. This API translates the code written by the end user (in Python, R of JavaScript) to a uniform JSON-format, which in its turn can make requests to different cloud computing back-ends for retrieving and/or processing remote sensing data. For this dissertation, Python is used as scripting language combined with the cloud back-end provided by VITO. Only the landuse data (cf. Section 2.4.1) could not be obtained through the openEO platform.

2.4.1 Land use data

As will be described in more depth in Chapter 3 (cf. Section 3.1.2), radar its interaction with the ground surface is highly dependent on surface characteristics. Therefore, land use data are of great importance to assign the proper causes to the observed SAR data. Although the Zwalm catchment is mostly located in Flanders, a small upstream part is located in Wallonia. Therefore the following two maps are combined: the Flemish land use map from 2019 at 10 m resolution in BD72 as CRS (Departement Omgeving, 2023) and the Walloon one from 2018 at 1 m resolution in Belgian Lambert 2008 (Service public de Wallonie, 2022). The reprojection, resampling and reclassifying to one map with five land use classes (urban, forest, pasture, agriculture and water) at 10 m resolution in BD72 was performed by other researchers at H-CEL. The final map used, clipped and masked to the Zwalm catchment, is given in Figure 2.7.

2.4.2 Vegetation data: PROBA-V and Sentinel-3

Because of the effect of surface characteristics on SAR data, it is of interest to have an indicator of how landcover by vegetation biomass changes over time. For this purpose, the leaf area index (LAI), being the total intercepting leaf area per unit horizontal ground area (Chen and Black, 1992), is chosen as the monitored biophysical variable. LAI retrieved via RS is chosen, more specifically the green LAI 300 m resolution product provided by Copernicus (2023). The product is derived from the PROBA-V satellite from January 2014 to June 2020 and from the Sentinel-3 satellites from July 2020 onward. The LAI is calculated on a 10-day basis and is made available in near-real time (Wolfs et al., 2022).

Since the SAR data will be spatially averaged out per land use category (cf. Section 3.3), this averaging will also be applied on the LAI data. For this purpose, the land

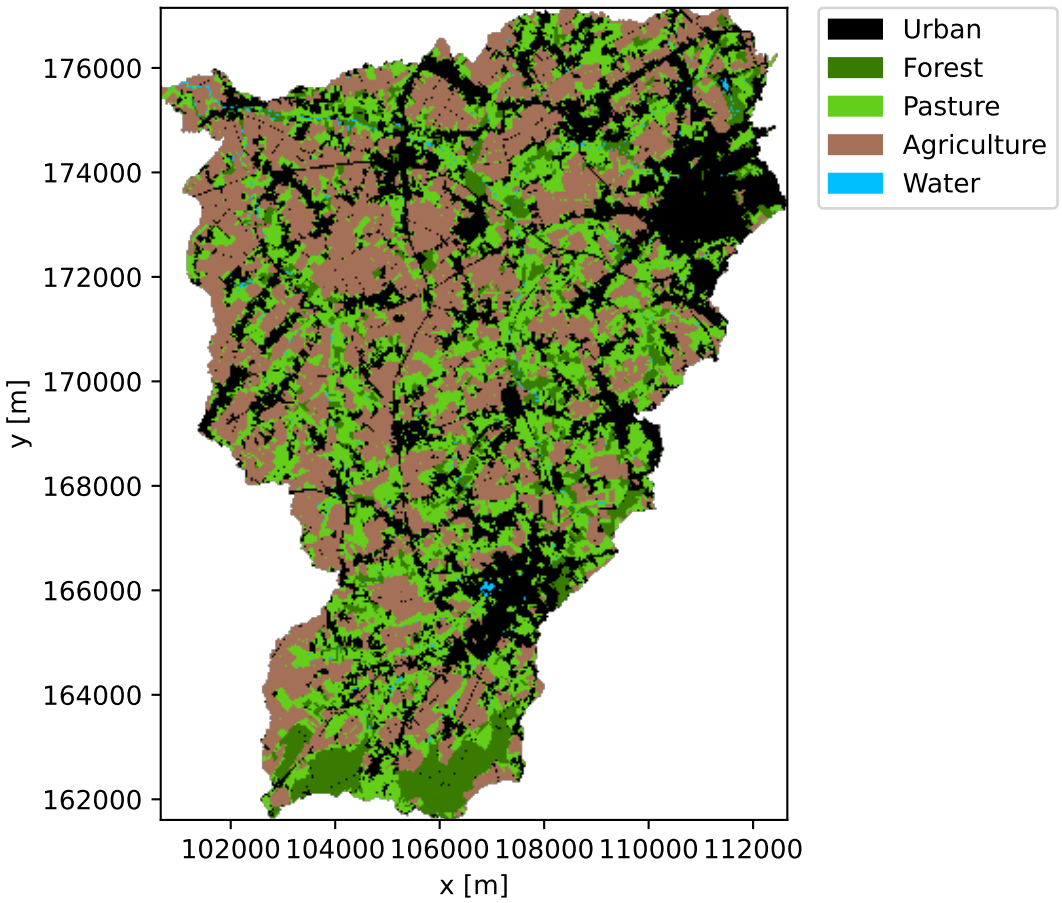

Figure 2.7: Land use data as adapted from Departement Omgeving (2023) and Service public de Wallonie (2022).

use data are first reprojected from from BD72 to WGS84 (the CRS for the LAI) with nearest neighbour resampling whilst keeping the original number of pixels. Next, the resolution is changed from 10 m to 300 m (the resolution of the LAI product) by taking the most frequently occurring land use category as the resampled value.

For reasons such as snow, ice or cloud cover, data can be lacking for certain pixels and/or time steps (Wolfs et al., 2022). To obtain a reliable LAI per land use category, it is chosen that at least 40% of the pixels belonging to that category needs to have a valid value to calculate the spatial mean at a certain time step.

Since the LAI data will be used as input to the ML methods together with the SAR data in Chapter 5, both should have data at the same points in time. Temporal resolution for these two datasets is not the same however: 10-daily resolution for LAI while the Sentinel-1 satellites provide data approximately every 3 days (cf. Section 3.2). Therefore, it it chosen to interpolate the LAI values. For this purpose, a piecewise cubic hermite interpolating polynomial (PCHIP) is used, as it preserves monotonicity between datapoints and does not overshoot non-smooth data. The interpolation method is based on applying a piecewise cubic polynomial between

each pair of known points. A more detailed description of the algorithm is omitted here, but can be found in Moler (2004).

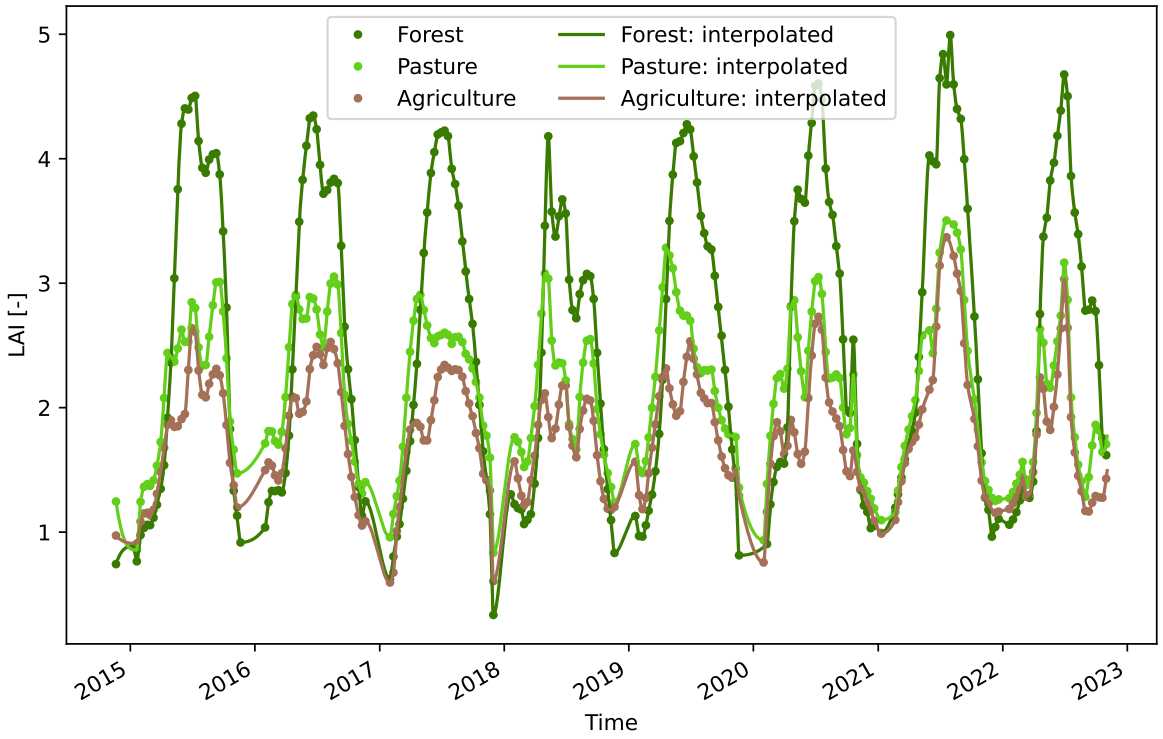

Figure 2.8: LAI time series per landuse category: original values as dots and interpolation as full lines.

The LAI time series averaged per land use category is displayed in Figure 2.8, where the dots represent the original data and the lines the PCHIP-interpolated data. Note that LAI data for the categories urban and water are omitted, as in these cases the retrieved LAI values have no physically meaningful interpretation. As expected, a clear seasonal pattern is present in the time series with LAI values high in the summer and low in the winter. Again according to expectation, forests reach higher LAI values in the summer than pastures or agricultural fields.

2.4.3 Radar data: Sentinel-1

A last, important category of satellite data is obtained from the Sentinel-1 SAR satellites. As SAR data products are not straightforward to interpret compared to the previously described remote sensing products, a separate chapter (cf. Chapter 3) is devoted to explaining the use and preprocessing of these data.

3. SAR: FROM SENTINEL 1 TO FEATURES

Heb sectie 3.1 over de theoretische achtergrond proberen in te korten t.o.v. de vorige versie. Niko las die versie al eens na. @Hans lukt het voor jou om dit deel eens na te lezen?

Before using the time series of satellite data coming from Sentinel-1 (S-1) for data-driven modelling (as will be described in Chapter 5), it is important to have a basic understanding of how these images are obtained. Therefore, this chapter will start by covering the basics of radar remote sensing (cf. Section 3.1) before describing the S-1 satellites (cf. Section 3.2) and how the data from this platform are processed to useful inputs (also called features, cf. Section 3.3) for data-driven modelling.

3.1 Theoretical background on radar remote sensing

Radar systems can be classified as active microwave remote sensing platforms. These active sensors send out bursts of electromagnetic radiation (EMR) and measure the reflected response from the Earth's surface (Woodhouse, 2006a). In the case of radar, the EMR is in the microwave spectrum, with wavelengths (λ) varying between 1 mm and 1 m. The use of these wavelengths has the following unique features/advantages (Woodhouse, 2006a; Lillesand et al., 2015):

- Microwaves can penetrate the atmosphere even when clouds are present, entailing an all-weather sensing capability. Combined with the day and night imaging of active systems, continuous coverage of the environment can be provided.
- With wavelengths different from those in the visible spectrum, microwaves provide an uniquely different view on the environment.

Note that microwave spectrum is further subdivided according to wavelengths, with the different bands indicated by a distinct letter. For synthetic aperture radar (SAR), the type of radar used in this dissertation, common bands are: X-band (2.4 - 3.75 cm), C-band (3.75 - 7.5 cm) and L-band (15-30 cm). Besides wavelength, EMR is also characterised by its polarisation, which describes the plane of oscillation for the electrical field. Oscillations occurring in the plane perpendicular to that of the surface imaged are called vertically polarised (V), whilst waves parallel to this

same plane are horizontally polarised (H). Depending on transmitting and receiving polarisation of the EMR respectively, four combinations can be used: VV, HH, HV or VH (Lillesand et al., 2015; Alaska Satellite Facility, nda).

3.1.1 From SLAR to SAR

To understand why SAR came to be, it is interesting to look at its simpler predecessor: the side-looking airborne radar (SLAR). The operational configuration of such a system is given in Figure 3.1, which shows an airborne mounted radar moving along in the azimuth direction while the radar is looking in the so called slant range direction. One of the most distinct features of S(L)AR operation is also illustrated: its side-looking nature, which is in contrast with the nadir oriented optical platforms. Other important terminology, as seen on Figure 3.1, is listed below (Meyer, 2019):

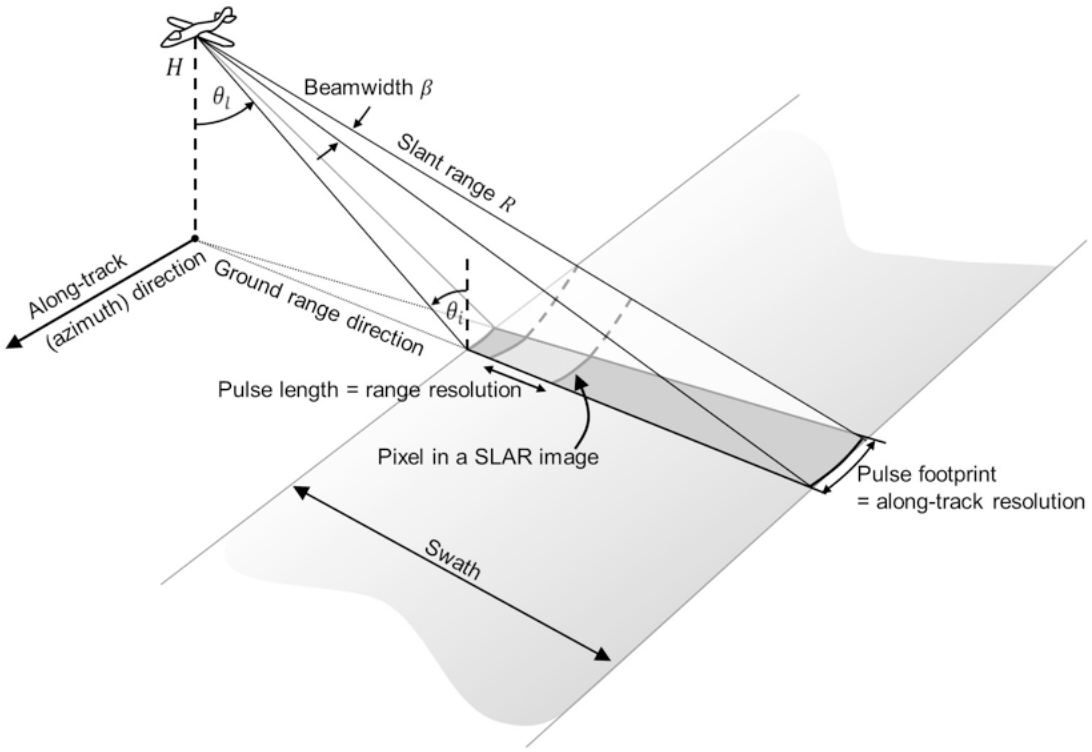

Figure 3.1: Example of SLAR operation (Meyer, 2019).

- Slant range = $R = \frac{ct}{2}$: distance from sensor to the detected object on the ground, where c is the speed of light and t the time between pulse transmission and receiving of the reflection.
- Swath: the area 'seen' by the sensor, usually defined by its width (Canada Centre for Remote Sensing, 2016)

Special attention should also be given to the different angles used to describe the radar geometry, which are also visualised in Figure 3.1 (Lillesand et al., 2015):

- Look angle = θ_l: angle between nadir and imaged point. θ_l will increase in the direction of the far swath. Note that on Figure 3.1, θ_l of the near swath is visualised.
- Incident (or incidence) angle = θ_i: angle between radar beam and the normal to the imaged area. When the local terrain at the point of incidence is taken into account, the term local incidence angle ($\theta_{i,l}$) is used.

Another important aspect of S(L)AR is the spatial resolution. For SLAR, the resolution in the slant range (ρ_R) is independent of R. The azimuth resolution (ρaz) on the other hand deteriorates with increasing R. As a consequence, R is too large for ρ_{az} to be useful in spaceborne applications. In SAR, this problem is alleviated by synthesising the effect of a very long antenna through the use of modified data post-processing and recording techniques. This is illustrated in Figure 3.2, with the synthesised antenna length (L_{SA}) being approximately equal to length of the flight path between the first (x_1) and last (x_2) observation of a certain object on the Earth's surface P (Lillesand et al., 2015; Meyer, 2019). For a more detailed and mathematically elaborate explanation of SAR, including a perspective based on Doppler shift, the reader is referred to Woodhouse (2006b)

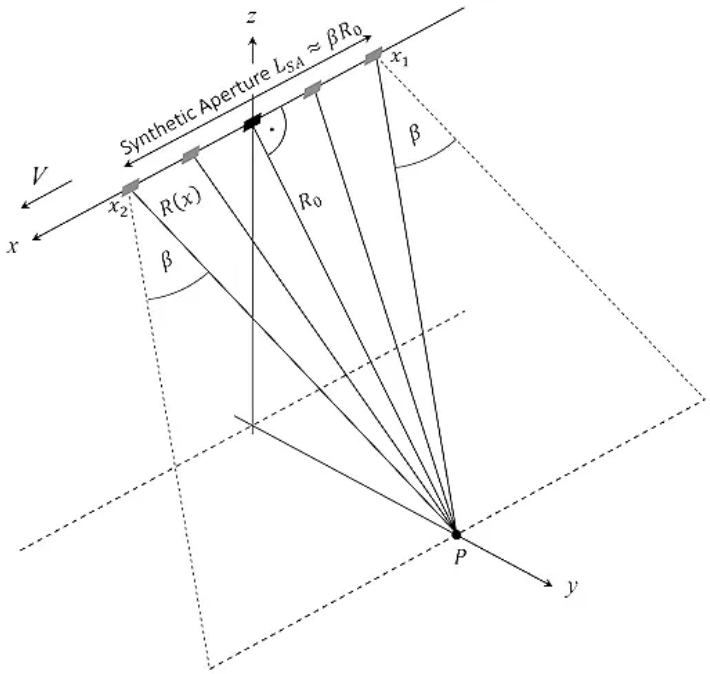

Figure 3.2: Simplified illustration of the concept of synthetic aperture radar (Meyer, 2019).

3.1.2 Key (environmental) factors affecting backscatter

3.1.2.1 Backscatter

Scattering is defined as the redirection of incident EMR by an object (Woodhouse, 2006c). In the case of an active sensor like for SAR, the key property to quantify for constructing a 2D image is the amount of transmitted energy that is returned back to the sensor. Therefore, the radar backscatter is defined as:

$$\beta = \frac{P_s}{P_i} \tag{3.1}$$

Remark that this β is not to be confused with the beamwidth of the antenna as visualised in Figure 3.1. P_s is the scattered power and P_i the incident power reaching the ground (Small, 2011). Note that despite not being explicitly stated as such by Small (2011), it seems recommended to define P_i as the power of the original incident wave in W/m^2 (as done by Woodhouse (2006c)) for consistency of units: with P_s in W, this yields a β in m^2, which in its turn gives unitless normalised backscatters.

Backscatter is not straightforward to interpret, as it is influenced by both sensor and landscape characteristics (Meyer, 2019). Furthermore, β also needs to be normalised by a reference area, as the backscattered power would otherwise increase with the measured area for distributed targets such as bare ground (Woodhouse, 2006c). Three different reference surface areas can be chosen (as visualised in Figure 3.3, with θ equivalent to θ_i or $\theta_{i,l}$) (Small, 2011):

- Radar brightness or *beta nought*:

$$\beta^0 = \frac{\beta}{A_\beta} \tag{3.2}$$

  Note that β^0 is the only parameter the radar system can measure directly, as the pixel area in the slant plain is known: $A_\beta = \rho_r \rho_{az}$ with $\rho_r = \delta_r$ and $\rho_{az} = \delta_a$ on Figure 3.3.

- Normalised radar cross-section or *sigma nought*:

$$\sigma^0 = \frac{\beta}{A_\sigma} = \beta^0 \sin(\theta_i) \tag{3.3}$$

  with A_σ the ground area. Without local topographic info available, this area is calculated based on an ellipsoidal model of the Earth. For remote sensing, it is a more interesting parameter than β_0 as it quantifies ground surface properties.

- Slope-normalised radar cross section or *gamma nought*:

$$\gamma^0 = \frac{\beta}{A_\gamma} = \beta^0 \tan(\theta_i) \tag{3.4}$$

  A_γ is the ground area projected in the plane perpendicular to the look direction. The main merit of using γ^0 is to remove the dependency on the incidence angle in the area used for normalisation (Woodhouse, 2006c).

As defined by Equations 3.2 to 3.4, the normalised backscatter coefficients are unitless and in a linear scale. For practical use however, these coefficients are often transformed to a decibel (dB) scale to allow for better visualisation and communication. For example for σ_0 this transformation is performed as:

$$\sigma^0\ [\text{dB}] = 10 \log_{10}(\sigma^0\ [\text{-}]\ ) \tag{3.5}$$

but this is completely analogous for β^0 and γ^0. Note however that for calculations, one should work in the linear scale (Dries and Van Tricht, 2019).

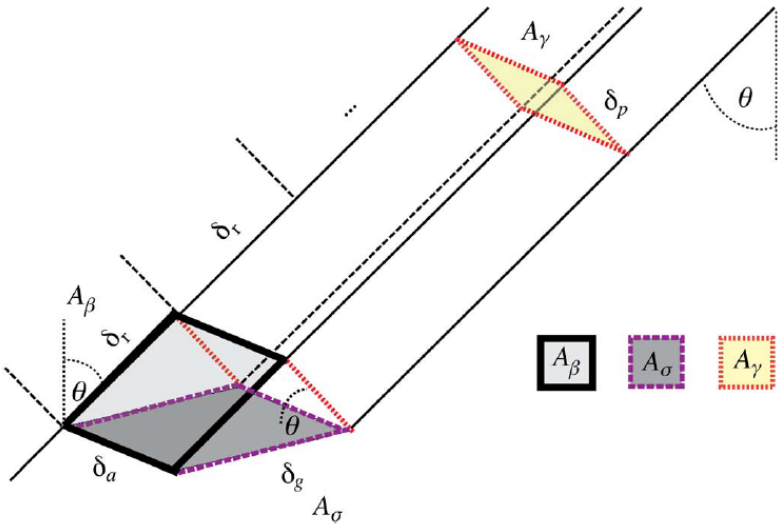

Figure 3.3: The three different areas for normalising SAR backscatter (Small, 2011).

3.1.2.2 Surface roughness

Surface roughness of the terrain determines if the surface can be seen as a specular reflector, which acts like a mirror and gives almost no return to the sensor, or a diffuse reflector, scattering radiation in almost all directions (including the one to the sensor). Based on the root-mean-square (rms) of the surface height variations, the classification based on the modified Rayleigh criterion can be made: rough (i.e.

a diffuse reflector) if rms > $\frac{\lambda}{4.4\cos(\theta_i)}$ and smooth (i.e. a specular reflector) if rms < $\frac{\lambda}{25\cos(\theta_i)}$ (Lillesand et al., 2015).

3.1.2.3 Electrical characteristics

The shorter the wavelength of the EMR, the less penetration (e.g. in the canopy or soil) will occur. For example, L-band microwaves will penetrate the canopy and ground, while X-band would reflect before reaching the ground (Alaska Satellite Facility, nda). This is an important characteristic, as the data-driven modelling of Chapter 5 has the goal of inferring information on soil properties.

For a bare surface, the penetration depth of the EMR is approximated as (Meyer, 2019):

$$\delta_p \approx \frac{\lambda\sqrt{\epsilon_r^{'}(\omega)}}{2\pi\epsilon_r^{''}(\omega)} \tag{3.6}$$

δ_p is dependent on the dielectric properties of the soil. These are described by its dielectric permittivity ϵ, which is denoted by ϵ_r when defined relative to to the permittivity of vacuum ϵ_0. As the EMR is sinusoidal time-dependent, ϵ_r is described by a complex number :

$$\epsilon_r(\omega) = \epsilon_r^{'}(\omega) - i\epsilon_r^{''}(\omega) \tag{3.7}$$

The real part $\epsilon_r(\omega)^{'}$ is called the dielectric constant and imaginary part $\epsilon_r(\omega)^{''}$ the loss factor (El Khaled et al., 2016).

$\epsilon_r^{'}(\omega)$ and $\epsilon_r^{''}(\omega)$ are functions of the frequency of the EMR $\omega = c/\lambda$. As ω increases (and thus λ decreases), $\epsilon_r^{'}(\omega)$ reduces while $\epsilon_r^{''}(\omega)$ increases, which when applied in Equation 3.7 results in the aforementioned reduced penetration depth for waves of low wavelength. More importantly however, both $\epsilon_r^{'}(\omega)$ and $\epsilon_r^{''}(\omega)$ are strongly influenced by soil moisture (SM), with higher SM leading to an increased value of both (Meyer, 2019). Note that for a dry soil $\epsilon_r^{'}(\omega)$ is around 4, while it is around 80 for a saturated soil. Despite the slightly faster increase of $\epsilon_r(\omega)^{'}$ compared to $\epsilon_r^{''}(\omega)$ when SM levels rise, $\sqrt{\epsilon_r^{'}(\omega)}/\epsilon_r^{''}(\omega)$ will still decrease so that Equation 3.7 will yield a lower δ_p for a higher SM content. Consequently, an increased radar reflectivity will be observed (Woodhouse, 2006c). Important to note here is that the C-band signal will only be indicative of the top few centimetres of the soil and the corresponding surface soil moisture (SSM) (Owe and Van de Griend, 1998).

3.1.2.4 Vegetation and urban environments

Besides the earlier mentioned diffuse and specular reflector, two other important type of scatteres can be defined (Meyer, 2019):

1. Volume scatterers are characterised by the multiple bouncing occurring within the structure. A vegetation canopy is the most common example.
2. Double-bounce scatterers have two distinct bounces that reflect the incident EMR right back to the sensor. These are vertical structures like buildings or tree trunks.

3.1.2.5 Influence of polarisation

Depending on polarisation of the EMR, the scattering types above will contribute to a varying extent to the observed backscatter. When VV polarisation is used, it is more sensitive to rough surface scattering (diffuse relfector), while HH has a greater sensitivity to double-bounce scattering. Concretely, this entails that when using double polarisation, one can distinguish between areas where double-scattering (high HH backscatter) or rough surface scattering (high VV backscatter) is predominant (Meyer, 2019).

3.1.3 Radar image characteristics

Besides the environmental influences on the backscatter discussed in the previous section, there are two other radar image characteristics to consider. The first is related to the viewing geometry. Because of the side-looking nature of SAR, geometric distortions occur compared to how geographic features occur in the ground range. This is especially of importance in hilly terrain, as mountain tops have slant ranges shorter than an object at the same coordinates with no elevation would. Additionally, a mountain can also block the sensor from detecting surface area behind the mountain (from the sensor's viewpoint) (Alaska Satellite Facility, nda).

The second important image characteristic is radiometric. In radar images, seemingly random variations in brightness occur which are called speckle. These variations are caused by the constructive and destructive interference of microwaves once backscattered, as the geometry within one pixel can slightly alter the distance from the antenna for each wave and therefore bring them out of phase. As these variations are not indicative of the true difference in backscatter between pixels, it is an unwanted image characteristic (Lillesand et al., 2015).

3.2 Sentinel-1

3.2.1 General information on satellite properties

The Sentinel-1 mission is part of Copernicus, the European Union's Earth observation programme (Copernicus, nd). The mission consists of two C-band SAR (ω =

5.405 GHz or equivalently $\lambda \approx 5.54$ cm), near-polar, sun-synchronous satellites called Sentinel-1A (S-1A) and Sentinel-1B (S-1B), which both have a 12 day repeat cycle. Consequently, the two satellites combined have a global exact revisit every 6 days. Due to overlapping swaths of different orbits however, a shorter revisit frequency of only around 3 days is achieved at the equator (and even shorter revisits for more northern latitudes). As the satellites are in sun-synchronous orbit, they pass at a fixed local solar time for every location: 18:00 when ascending and 06:00 when descending. S-1A and S-1B were launched on 03/04/2014 and 25/04/2016 respectively with a provisioned operational lifespan of 7 years, after which they will be replaced by S-1C and S-1D. Unfortunately, the operational lifespan of S-1B ended prematurely in July 2022 (ESA, nda).

The S-1 satellites have four different acquisition modes: Stripmap, Interferometric Wide Swath (IW), Extra Wide Swath and Wave. As IW is the primary operational mode over land, it is also the one used in this dissertation. For this acquisition mode, both single (HH or VV) and dual (HH+HV or VV+VH) polarisation are supported. IW has a 250 km swath width, which is formed by the processing of data from three subswaths (ESA, ndb).

3.2.2 Data products

Depending on the level of processing applied, three different levels of Sentinel-1 data are generated: Level-0, Level-1 and Level-2. As Section 3.3 starts from Level-1 data, only these first two levels will be covered below.

Level-0 Level-0 SAR products are the raw data. It is the basis for all the other products that can be derived. By the end-user, this format is rarely used however as the European Space Agency (ESA) processes these products to Level-1 data with their Instrument Processing Facility (ESA, ndb). Within 24 hours of acquisition, this pre-processing is completed and the data can be obtained from the Copernicus Open Acces Hub (Alaska Satellite Facility, ndb).

Level-1 Depending on the processing steps applied to Level-0 data, two different Level-1 products can be obtained: single look complex (SLC) or ground range detected (GRD) (ESA, ndb):

- SLC: Provides SAR data in the natural slant range geometry. By using a complex value for each grid cell, both amplitude and phase information are preserved. Furthermore, it has a pixel spacing of 2.3 x 14.1 m (range x azimuth) for IW acquisition.

- GRD: The SAR data has been projected to ground range by using the WGS84 Earth ellipsoid model. Only amplitude and no phase information is preserved in this format. The combination of multilooking (averaging the information over several pixels, in this case 5 range pixels by 1 azimuth pixel) and projecting to ground range results in a 10 x 10 m pixel spacing at the highest resolution for IW.

3.3 Pre-processing

3.3.1 GRD to σ^0 or γ^0_T processing

As explained in Section 3.1.3, SAR imagery is characterised by both geometric and radiometric distortions. Furthermore, the imagery also needs to be calibrated to one of the options described in Section 3.1.2: β^0, σ^0 or γ^0. The current state-of-the-art method is first processing to β^0 and then applying radiometric terrain flattening to a terrain-flattened γ^0 (denoted as γ^0_T) with the method described in Small (2011). This method improves the representation of radar brightness in hilly areas, as not only geometry but also radiometry is influenced by terrain variations. Unfortunately, γ^0_T data are not readily available and the end user has to perform the conversion from GRD to γ^0_T themselves. σ^0 data on the other hand are processed and distributed by (VITO, nd) for Belgium. To arrive at the final, σ^0 product, following steps are undertaken at VITO using SNAP and GDAL (Dries and Van Tricht, 2019):

1. Apply orbit correction: update the orbital state vectors (i.e. position, velocity and time) with the information available only after acquisition to improve geolocation accuracy.
2. GRD border noise removal: In the generation of Level-1 GRD-products, radiometric artefacts are created at the borders of the imagery. This step removes that type of noise.
3. Thermal noise removal: remove noise coming from electronic instruments.
4. Radiometric calibration to σ^0: cf. Section 3.1.2.1 for the importance of and methods for radiometric calibration.
5. Range Doppler terrain correction: orthorectification to account for the the geometric dislocation of pixels (cf. Section 3.1.3) using a DEM. For this correction the WGS84-ellipsoid and CRS Universal Transverse Mercator (UTM) zone 31N are used.
6. Finalising GDAL processing steps to Cloud Optimized Geotiffs.

Note that in the above processing chain, no speckle removing filter is applied. The data are retrieved via the openEO Platform with the VITO backend (cf. Section 2.4).

For the own processing of GRD to γ_T^0 via the openEO platform (again with the VITO backend), a very similar processing chain (with regard to the first five steps) is executed. The major difference lies in the radiometric calibration, as the GRD products are first calibrated to β^0 before being corrected to:

$$\gamma_T^0 = \beta^0 \frac{A_\beta}{\int_{DEM} A_\gamma} \tag{3.8}$$

where the area in the denominator is obtained via integration over the DEM, which is more accurate than the (local) incidence angle based approach used in Equation 3.4 (Small, 2011). For this purpose (but also for the orthorectification), the Copernicus Global 30 meter DEM dataset (Copernicus, 2022) is used. The full description of the processing chain executed by the Sentinel Hub cloud computing platform, which is accessed by the VITO backend, is given by Sentinel Hub (2023). Note that the final product is in compliance with the CEOS Analysis Ready Data For Land (CARD4L) specifications on normalised radar backscatter (Bontje et al., 2022; CEOS, nd; openEO Platform, 2023).

Both σ^0 and γ_T^0 data are retrieved at 10 m pixel spacing and in VV and VH polarisations, as both are provided over land in IW acquisition mode (ESA, ndb) and reflect different scattering behaviours depending on the terrain (cf. Section 3.1.2.5). The data are available starting from 07/06/2015 and consulted until 05/11/2015. Furthermore, only days when the SAR image covers the full catchment are included for further analysis.

3.3.2 Spatial averaging of σ^0 or γ_T^0 data

Based on the interactions of microwave EMR with the environment, it is clear that not for all land use categories the observed backscatter will be related to SM variations. This is especially the case for the following land uses:

- Urban: high backscatter values are expected due to the abundance of double-bounce scatters (cf. Section 3.1.2.4) associated with buildings. These values are however not indicative of a high SM content. For illustrative purposes, one σ^0 calibrated image, taken on 10/06/2015, is visualised together with the land use map in Figure 3.4. Note that the land use data are reprojected to the CRS of the backscatter data (UTM zone 31N) with nearest neighbour resampling for visualisation. In the upper right corner of this figure, the correlation between urban landuse and high σ^0 values is apparent.
- Forest: In the canopy of forest, volume scattering will occur (cf. Section 3.1.2.4). Consequently, C-band radar will have more difficulties penetrating to the ground surface (Alaska Satellite Facility, nda), which is required for retrieval of SM information.

- Water: With its relatively smooth surface, open water surfaces are typically specular reflectors (cf. Section 3.1.2.2). This scattering mechanism will result in low backscatter values (Landuyt et al., 2021). Once more, the physics governing this interaction are not related to the SM content.

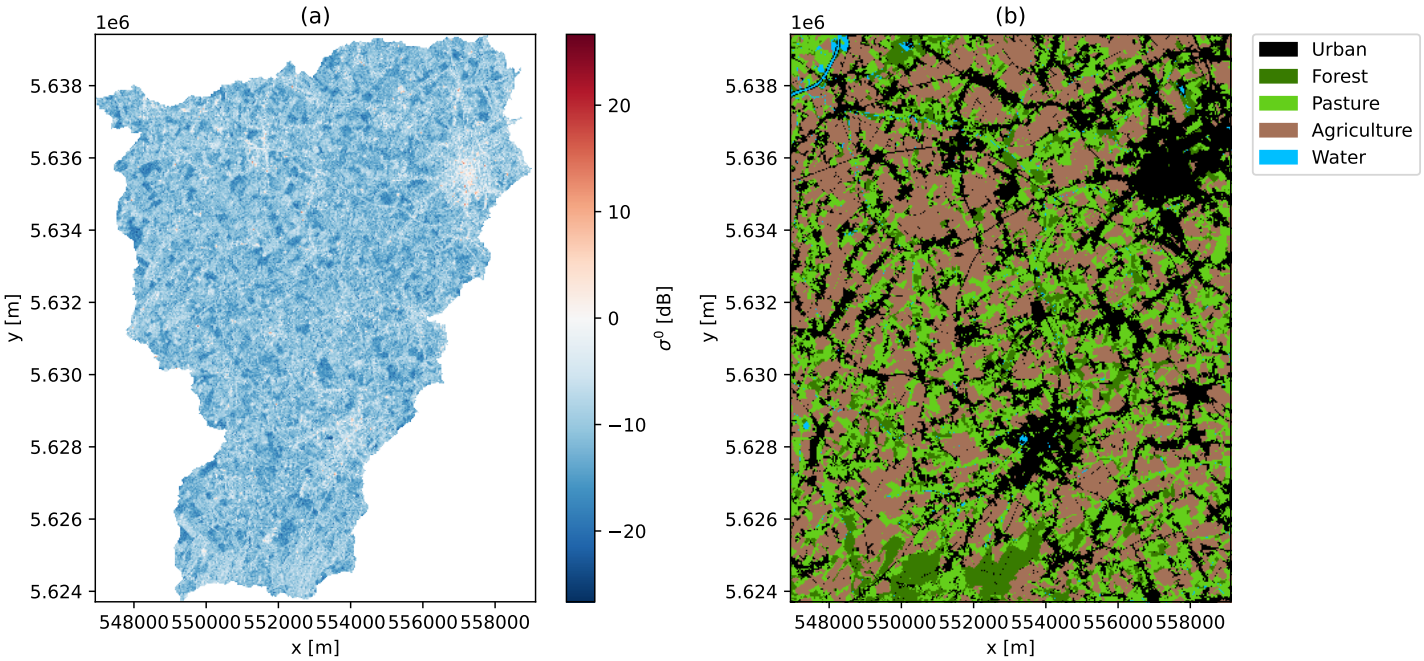

Figure 3.4: (a) σ^0 calibrated SAR backscatter over the Zwalm on 10/06/2015 (b) Land use as described in Section 2.4.1.

Based on the physical principles of SAR, it is therefore hypothesised that mostly backscatter averaged over the land use categories pasture and agriculture will provide relevant information on SM and hence the state variable to be estimated. The effect of whether or not to include the forest averaged time series combined with the agriculture and pasture backscatter as inputs for the ML model, will be investigated in Chapter 5. Note that for these three land uses, LAI data are also available (cf. Section 2.4.2).

The time series after spatial averaging for the three land uses of interest are displayed in Figure 3.5. The ascending and descending orbit are displayed separately, as the differing viewing geometries impact the computed backscatter. Note that for all time series, some seasonality can be observed in the pattern with higher backscatter in winter than summer, which is likely to be related to elevated SM levels (cf. Section 3.1.2.3). This pattern is generally less noisy for the ascending orbit than for the descending one. Of the three landuse categories, most seasonal variation is exhibited by the backscatter averaged over agricultural areas.

An analogous figure to the above is given for σ^0 with Figure A.1. In general, it can be concluded that the σ^0 time series does not show trends drastically different from those for γ^0_T. This is also confirmed when the difference between the calibrated time series (splitted up according to the four categories of Figure 3.5) are examined

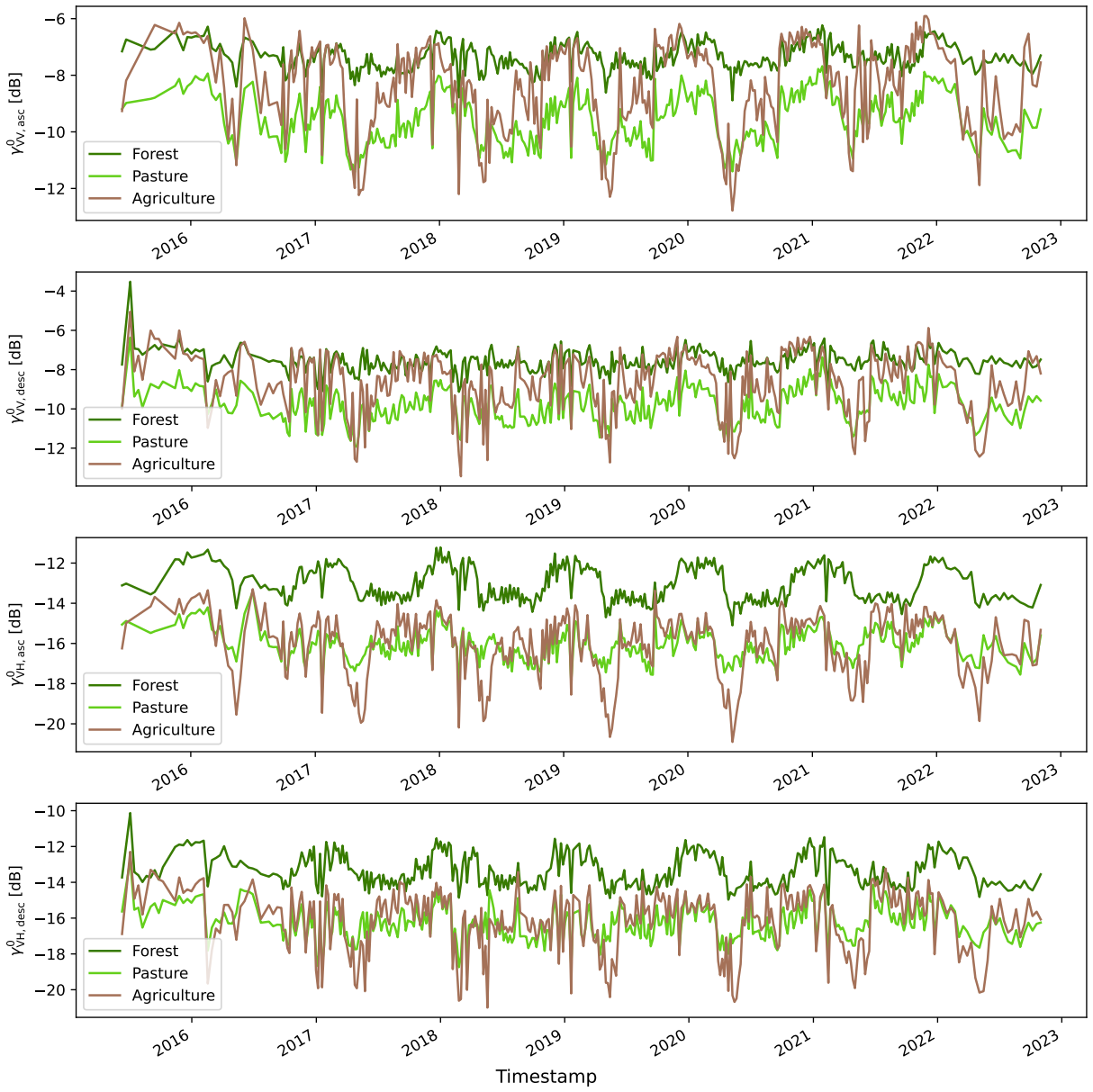

Figure 3.5: Time series for γ^0_T in (from top to bottom) VV with ascending orbit, VV with descending orbit, VH with ascending orbit and VH with descending orbit.

(see Figure A.2), as a fairly constant difference in the range of 1 dB between the two is mostly observed. The difference is the most jagged for forest averaging and less so for pasture and agriculture. In what follows, γ^0_T will be used as it is the CARD4L recommend calibration for radar backscatter (Bontje et al., 2022). It is however hypothesised that due to the quite constant difference between the two time series combined with the relatively flat Flemish terrain, use of σ^0 data as input for the ML inverse observation operator would also be suitable.

4. THE PROBABILITY DISTRIBUTED MODEL (PDM)

Dit stuk is door Niko al nagelezen en kleine veranderingen uit de opmerkingen zijn doorgevoerd. Minder noodzakelijk denk ik dat één van jullie 2 dit opnieuw naleest.

4.1 Introduction to rainfall-runoff modelling

Hydrological rainfall-runoff models can be classified in many ways, but the following non-exhaustive list of criteria is often used (Solomatine and Wagener, 2011; Beven, 2012b):

- Deterministic vs. stochastic models, where the latter considers randomness i.e. a given input does not always give the same output.
- Spatial representation: lumped models consider the catchment as a single homogeneous unit, while distributed models considers multiple cells within the catchment.
- Model structure and parameters:
  - Conceptual: structure determined a priori, parameter calibrated on empirical data
  - Physically based: parameters and structure determined a priori
  - Data-driven: parameters and structure based on empirical data

In this dissertation the focus will be on deterministic, lumped, conceptual models, as these are often used in operational forecasting (Solomatine and Wagener, 2011).

A large number of these conceptual models, sometimes called explicit soil moisture accounting models, exist (Beven, 2012a). Examples are the HBV (or HBV-96 for the updated version) model, which can be lumped or semi-distributed (i.e. the lumped structure is applied for multiple sub-basins) and is often used in Scandinavia (Lindström et al., 1997), and the GR4J-model, which is a lumped four-parameter model developed in France (Perrin et al., 2003). In this dissertation, the focus will be on the probability distributed model (PDM) (Moore, 2007), as it is commonly used in Flanders, amongst others by the Flemish Environment Agency (VMM) who uses

this model in their real-time flood forecasting system for the unnavigable rivers (Dewelde et al., 2014).

The PDM itself will be described in more detail in Section 4.2 and its necessary calibration in Section 4.3. It is interesting to note that this calibration procedure is an ill-posed optimisation problem, as not enough information is available in the used data sets to find one robust parameter optimum. This leads to the concept of equifinality, which embraces that due to shortcomings in both model structure and observed data, multiple model structures and parameter sets can be valid simulators of observed hydrological behaviour (Beven, 2012b).

4.2 Description of the PDM

The description given below is entirely based on Moore (2007), to which the reader is referred for a more in-depth mathematical explanation. The general model structure is given in Figure 4.1, with the inputs I and E equivalent to precipitation (P) and potential evaporation (PE) respectively and Q the flow at the catchment outlet (as obtained in Chapter 2). Three distinct reservoirs are used in the model structure and covered below.

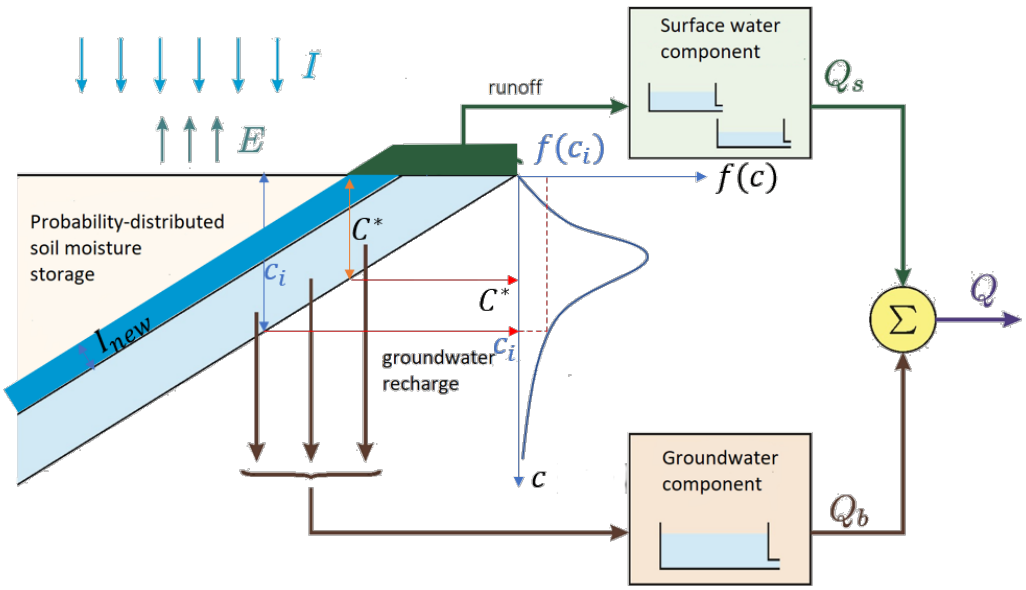

Figure 4.1: Overview of the PDM's model structure, adapted from Moore (2007).

Probability-distributed soil moisture storage S_1 Despite the lumped model structure, spatial variability in the soil moisture capacity c [mm] is described by assigning a probability density to c according to the probability density function (PDF) $f(c)$ (as illustrated for c_i in Figure 4.1). Different distributions can be used as PDF, but mostly the Pareto distribution is used which has the following cumulative

distribution function (CDF):

$$F(c) = 1 - \left(\frac{c_{max} - c}{c_{max} - c_{min}}\right)^b \text{ with } c_{min} \leq c \leq c_{max} \tag{4.1}$$

with c_{max} [mm] and c_{min} [mm] the maximum and minimum soil moisture storage capacities and b [-] a parameter controlling the spatial variability of c. For each time step, the critical capacity C^* is calculated. All storages with a capacity less than C^* will be generating runoff when rain falls. This is illustrated in Figure 4.1 for one time interval: with I_{new} the net rainfall and the updated C^*, the trapezoid in green above the storages with $c < C^*$ is the amount of runoff produced.

Out of $C^*(t)$, the total soil moisture storage $S_1 = S$ [mm] can be determined as:

$$S(t) = \int_0^{C^*(t)} cf(c)dc + C^*(t)\int_{C^*(t)}^{\infty} f(c)dc = \int_0^{C^*(t)} (1 - F(c))dc \tag{4.2}$$

where the notation based on $f(c)$ is related to the one with $F(c)$ via integration by parts. With $S(t)$ known, the actual evaporation (E) is determined from the potential evaporation (PE) as:

$$\frac{E_i}{PE_i} = 1 - \left(\frac{S_{max} - S(t)}{S_{max}}\right)^{b_e} \tag{4.3}$$

with S_{max} the expected value of c $\left(E(c) = S_{max} = \int_0^{\infty} cf(c)dc\right)$ and b_e an exponent that is usually set to 2. The i subscript is used to clarify that the corresponding variable/state belongs to the i^{th} time step ($t \longrightarrow t + \Delta t$ with $\Delta t = 1$ h). The groundwater recharge from the soil moisture storage d_i [mm/h] is expressed as:

$$d_i = k_g^{-1}(S(t) - S_t)^{b_g} \tag{4.4}$$

with S_t [mm] the threshold storage below which no drainage occurs, b_g an exponent usually set to 1 and k_g [h mm$^{b_g - 1}$] the groundwater recharge time constant. With the equations for E_i an d_i, the net rainfall π_i [mm/h] during the i^{th} time step is determined as:

$$\pi_i = P_i - E_i - d_i \tag{4.5}$$

which can be used to update C^*:

$$C^*(t + \Delta_t) = C^*(t) + \pi_i \Delta_t \tag{4.6}$$

Consequently, $S(t + \Delta t)$ can be calculated according to Equation 4.2, and the runoff produced, when $\pi_i > 0$ and $C^*(t + \Delta_t) > c_{min}$, as:

$$\begin{cases} V(t + \Delta t) = \pi_i \Delta_t - (S(t + \Delta t) - S(t)) & \text{if } S(t + \Delta_t) < S_{max} \\ V(t + \Delta t) = \pi_i \Delta_t - (S_{max} - S(t)) & \text{if } S(t + \Delta_t) = S_{max} \end{cases} \tag{4.7}$$

Surface water component S_2 S_2 is a fast reacting reservoir used to represent routing of surface runoff. It is composed of two linear reservoirs (i.e. the flow out of the reservoir is proportional to the storage in the reservoir) with time constants k_1 [h] and k_2 [h]. The first reservoir receives V as input which is coming from S_1 and the output of the second reservoir is the surface flow Q_s [mm/h].

Groundwater component S_3 The groundwater storage is slowly reacting and meant to represent baseflow routing. It is made up of one non-linear reservoir expressed as:

$$\frac{dS_3}{dt} = d_i - Q_b \text{ with } Q_b = k_b S_3^m \tag{4.8}$$

with k_b the baseflow time constant [$h^{-1}mm^{1-m}$] and Q_b [mm/h] the baseflow of component of the total flow.

Ultimately, the total basin flow Q [m^3/s] can be calculated as:

$$Q(t + t_d) = Q_s + Q_b + Q_c \tag{4.9}$$

where Q_b and Q_s are converted from mm/u to m^3/s by using the area of the catchment, Q_c is a constant flow accounting for any abstractions (e.g. drinking water production) or additions to the catchment flow and t_d [h] a time delay factor which allows shifting the modelled hydrograph to better match the observed one. In this research, the PDM will be run at an internal time resolution (Δt) of 1 hour. The output Q however is averaged out to daily values.

4.3 Calibration

Despite that the PDM has already been calibrated for the Zwalm by the VMM (cf. Section 4.3.2), it will be recalibrated to gain better model performance over the considered period (2012-2023). As an objective function, the Nash-Sutcliffe efficiency (NSE) is used (Nash and Sutcliffe, 1970):

$$\text{NSE} = 1 - \frac{\sum_{k=1}^{N}(Q_{o,k} - Q_{m,k})^2}{\sum_{k=1}^{N}(Q_{o,k} - \bar{Q}_o)^2} = 1 - \frac{\sigma_e^2}{\sigma_o^2} \tag{4.10}$$

with N the number of considered time steps, the subscript k denoting the time step, $Q_{o,k}$ the observed flow, $\bar{Q}_o$ the average observed flow, $Q_{m,k}$ the modelled flow, σ_e^2 the error variance and σ_o^2 the observation variance. A NSE of 1 is a perfect model, while a NSE of 0 means that the model has the same predictive power as the mean of the observations. With NSE as an objective function, following limitations apply related to the squared error: more importance is given to predicting peak flow, high sensitivity to timing errors and biased parameters obtained when flows

residuals are autocorrelated in time (Beven, 2012c). Therefore, also other metrics will be used for model evaluation. The modified Nash-Sutcliffe efficiency (mNSE) is defined equivalently to the NSE but the square of flow differences is replaced by the absolute difference, resulting in a metric better suited for baseflow rating (Jiang et al., 2020; Legates and McCabe Jr., 1999). For rating peak flow on the other hand, percent bias in flow-duration curve high-segment volume (FHV) is well suited (Höge et al., 2022). FHV is defined as (Yilmaz et al., 2008):

$$\text{FHV} = 100 \frac{\sum_{h=1}^{H} Q_{m,h} - Q_{o,h}}{\sum_{h=1}^{H} Q_{o,h}} \tag{4.11}$$

with $h = 1, \ldots, H$ indicating the flows with an exceedance probability p smaller than 2%. These probabilities are defined as the complement of the empirical CDF ($p = 1 - P(Q < q)$) (Vogel and Fennessey, 1994) and are calculated for Q_o and Q_m separately.

Note that for all performance metrics, time steps with no values for Q_o are ignored. Furthermore, the data series is split up in two periods:

1. Calibration period: 01/01/2012 00:00 until 31/12/2019 23:00
2. Validation period: 01/01/2020 00:00 until 05/11/2022 23:00

For the calibration period a warm-up period of 9 months is taken, which is needed for the state variables to reach an optimal state and is excluded in the calculation of the performance metrics. For the validation period, optimal states are transferred from the calibration period and hence no warm-up is required. In what follows, the full period is defined as the end of warm-up until the end of validation.

4.3.1 Calibration algorithms

Many different automatic optimisation techniques exist, but the focus here will be on one example of two broad categories. The first (cf. Section 4.3.1.1) is a local method based on the idea of direct search. This means that starting from one initial parameter set, a 'hill-climbing' technique is applied that explores different trial directions to change the parameter set to a higher performance metric value. It differs from the gradient algorithms which determine the direction of change according to the gradient of the objective function to the parameters. The latter is less applied for calibration of hydrological models as the gradient can often not be determined analytically. Because of the many local optima in the response surface, local algorithms have to be started from multiple initial conditions to find the global optimum (Solomatine and Wagener, 2011; Beven, 2012c).

The second (cf. Section 4.3.1.2) is a global calibration algorithm based on evolutionary search. These heuristic methods do not rely on continuity or differentiability of

the objective function, making them well suited for calibration of hydrological models. Different variants exist, such as genetic (e.g. the often used shuffled complex evolution of Duan et al. (1993)) or ant colony algorithms, but all are based on a 'population' of parameter sets to explore the full parameter space for the global optimum (Tayfur, 2017).

4.3.1.1 Nelder-Mead algorithm

The Nelder-Mead (NM) method, as first described in Nelder and Mead (1965), aims at finding the optimum (classically defined as the minimum) of a function with n variables by constructing a simplex (i.e. the n-dimensional equivalent of a triangle) with $n + 1$ vertices in the parameter space. At each iteration, the vertex with the least desired function value (i.e. the minimum for the NSE) is replaced by applying one of four possible operations. The algorithm is illustrated for $n = 2$ in Figure 4.2. One iteration of the maximisation problem in this figure can be summarised as:

1. Calculate the objective function at each of the vertices and rank according to score. For the current example, p_{max} is the best score and p_{min} the worst.
2. Determine the centroid of the n best vertices (so exclude p_{min})
3. Reflect p_{min} with regard to the centroid resulting in p_r. If $p_{min} < p_r < p_{max}$, p_r is the new vertex replacing p_{min}.
4. If $p_r > p_{max}$, the reflection was looking in the right direction and an expansion is applied resulting in p_e. p_e and p_r are compared and the highest scoring one becomes the new vertex, while p_{min} is excluded.
5. If p_r is less than p_{min}, two contraction points are defined: p_c and p_c^*. The highest of these two becomes the new vertex if its value is higher than p_{min}.
6. Only if p_c and p_c^* are both less than p_{min}, is the simplex shrunk and the new vertex is the middle point of p_{min} and p_{max}.

The implementation from `scipy` (Virtanen et al., 2020) is used in this dissertation with the standard parameters for reflection, expansion, contraction and shrink as provided in Gao and Han (2012). Convergence is reached when the maximum distance in the parameter space and maximum difference in objective function between the best vertex and the others are both at most 0.001. The search space is constrained to the minimum and maximum parameter values as given in Table A.1. Lastly, the algorithm is initiated from 50 different initial positions by randomly sampling over a uniform distribution between minimum and maximum parameter values.

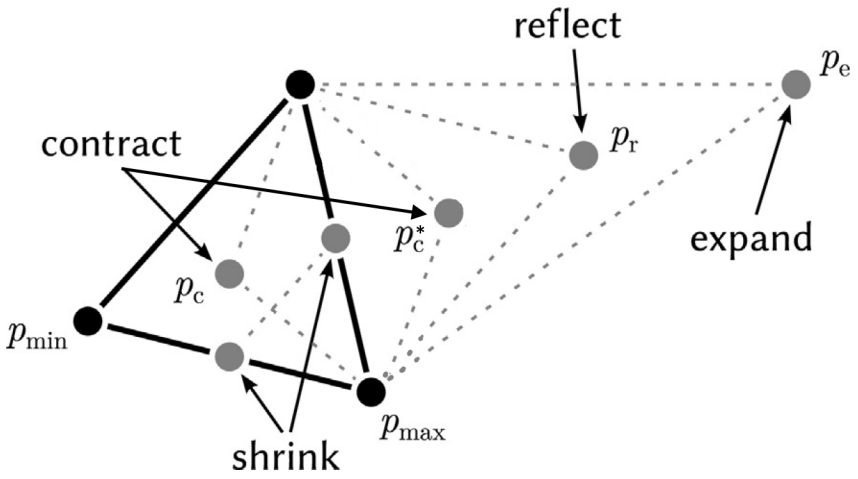

Figure 4.2: Illustration of the Nelder-Mead method applied to the two-dimensional maximisation problem, adapted from Cheng and Mailund (2015).

4.3.1.2 Particle swarm optimisation

The particle swarm optimisation (PSO) algorithm was first described by Kennedy and Eberhart (1995) and is a nature-inspired algorithm initially designed to simulate birds seeking food, for which individual birds interact with their neighbours (Shi, 2004). With this analogy in mind, each individual, evolving parameter set is called a particle. For each of the n_s particles, its position in the n-dimensional parameter space is updated as:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \tag{4.12}$$

with $\mathbf{x}_i(t)$ and $\mathbf{v}_i(t+1)$ the position and velocity vector of particle i at the t^{th} and $t^{\text{th}}+1$ discrete iteration step respectively. Furthermore $\mathbf{x}_i(0) \sim U(\mathbf{x}_{min}, \mathbf{x}_{max})$ with U a uniform distribution and $\mathbf{x}_{min}$ and $\mathbf{x}_{max}$ given in Table A.1. Global best (*gbest*) PSO with inertia weight is applied here for which the neighbourhood of each particle consists of all the other particles, which results in following velocity for particle i in dimension j (with $j \in \{1, \ldots, n\}$):

$$v_{ij}(t+1) = w v_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[\hat{y}_j(t) - x_{ij}(t)] \tag{4.13}$$

where following interpretations can be given to the right-hand side of the equation:

1. $v_{ij}(t)$: momentum component with w the inertia weight.
2. $c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)]$: cognitive component reflecting the particle's own memory of its best (i.e. best value of the objective function) location $\mathbf{y}_i(t)$ up until t. c_1 is an acceleration coefficient and $r_{1j}(t) \sim U(0,1)$ brings stochasticity in the algorithm.
3. $c_2 r_{2j}(t)[\hat{y}_j(t) - x_{ij}(t)]$: social component where each particle is drawn to the best position up until t of all particles $\hat{\mathbf{y}}(t)$. c_2 and $r_{2j}(t)$ are defined analogously as above.

Note that also local best PSO exists where the influencing neighbourhood of each particle is smaller than the total swarm. *gbest* is preferred here however because of it faster convergence (which comes at the cost of less exploratory behaviour) (Engelbrecht, 2007). Over the course of 50 iterations (after which the optimisation is terminated) (Piotrowski et al., 2020), w is linearly decreased from 0.9 to 0.4 (Shi and Eberhart, 1999), c_1 linearly increased from 1.3 to 2 and c_2 linearly decreased from 2 to 1.3 as applied in the calibration of the semi-distributed hydrological model HEC-HMS by Kamali et al. (2013). Although a number of empirical studies have shown successful calibration with relatively small swarm sizes ($n_s \in \{10, \ldots, 30\}$) (Engelbrecht, 2007), more recent research by Piotrowski et al. (2020) indicates that for most PSO variants (including inertia weight *gbest* PSO) swarm sizes in the range of 70 to 500 perform better. A swarm size of 150 is chosen as a compromise between the faster computation time of smaller swarms and greater exploratory ability of larger ones. The used implementation is from the `PySwarms` package (Miranda, 2018).

4.3.2 Initial parameter set

The parameter set as calibrated by the VMM is given in Table A.1. Note that the VMM uses a catchment area of only ±109 km^2, which is smaller than the ±115 km^2 from the delineation of Chapter 2. The parameter set was obtained by calibrating over events (so not the entire time series) in the period 07/1972 - 12/2001 (Cabus, 2021). The modelled flow time series ($Q_{m,init}$) is compared with the observed flow (Q_o) in Figure 4.3. The most apparent feature is the general overestimation of peak flows. This is also reflected in the high FHV given in Table 4.1. The performance over the validation period is also significantly better than over the calibration period for all three metrics. Over the full period, the NSE is slightly below the 0.5 threshold often used for an acceptable performance (Moriasi et al., 2015), confirming the need for further calibration.

Table 4.1: Performance metrics for initial, NM optimised and PSO optimised parameter sets for the PDM. Best values per period for NSE, mNSE and FHV are displayed in green, blue and bold respectively.

Period	**Initial**			**NM optimal**			**PSO optimal**		
	NSE	mNSE	FHV%	NSE	mNSE	FHV%	NSE	mNSE	FHV%
Calibration	0.2613	0.2744	49.54	0.7514	0.5096	**-7.65**	0.7210	0.5110	-18.41
Validation	0.7725	0.5525	**14.90**	0.7719	0.6233	-34.99	0.6830	0.5704	-44.25
Full	0.4708	0.3689	35.63	0.7601	0.5465	**-17.90**	0.7059	0.5312	-28.09

4.3.3 Results

For both algorithms, calibration was carried out with the catchment area of ±115 km^2 from Chapter 2.

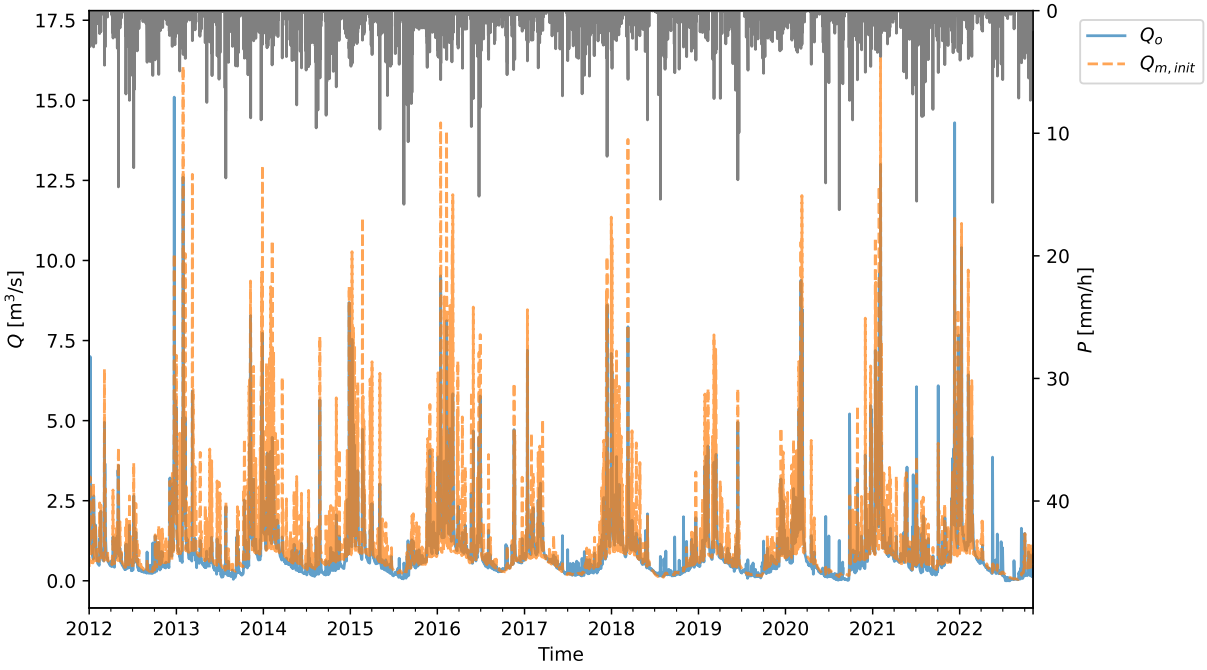

Figure 4.3: Observed Q_o compared to $Q_{m,init}$ predicted by the PDM. At the top of the figure, rainfall intensity is displayed.

4.3.3.1 Nelder-Mead algorithm

To compare the results of the 50 different initial positions, the dotty plots of all the different parameters are shown in Figure 4.4 with the NSE on the full period as metric. The full range of possible parameter values (cf. Table A.1) are shown on the x-axis. Only parameters with a NSE above 0.5 are displayed for clarity and because only three parameter sets have lower NSE's (which are even < 0). The concept of equifinality is clearly illustrated here, as for certain parameters (e.g. c_{min}, k_b and S_t) equivalently performant models are obtained over the entire possible range. For other parameters there are more clear trends in which ranges are suitable: better performance in the lower range for c_{max}, t_d and Q_c, while the higher range is preferred for b_e.

The best parameter set is selected based on the highest weighted sum of NSE and mNSE for the validation period, as this is deemed indicative of best generalisation on unseen data for both peak (∼ NSE) and base (∼ mNSE) flow. This parameter set is given in Table A.1 and its performance metrics in Table 4.1. For nearly all metrics, a substantial improvement is obtained over the initial parameter set. Only for the validation period an increased bias is observed for the FHV while the NSE stays practically the same. In general, peak flow underestimation occurs, as can be seen on Figure 4.5 (where $Q_{m,NM}$ denotes the modelled flow with the NM optimal set), as opposed to the earlier overestimation for the initial set. Also baseflow is still not always well simulated, as for example in the years 2013-2015 it is mostly

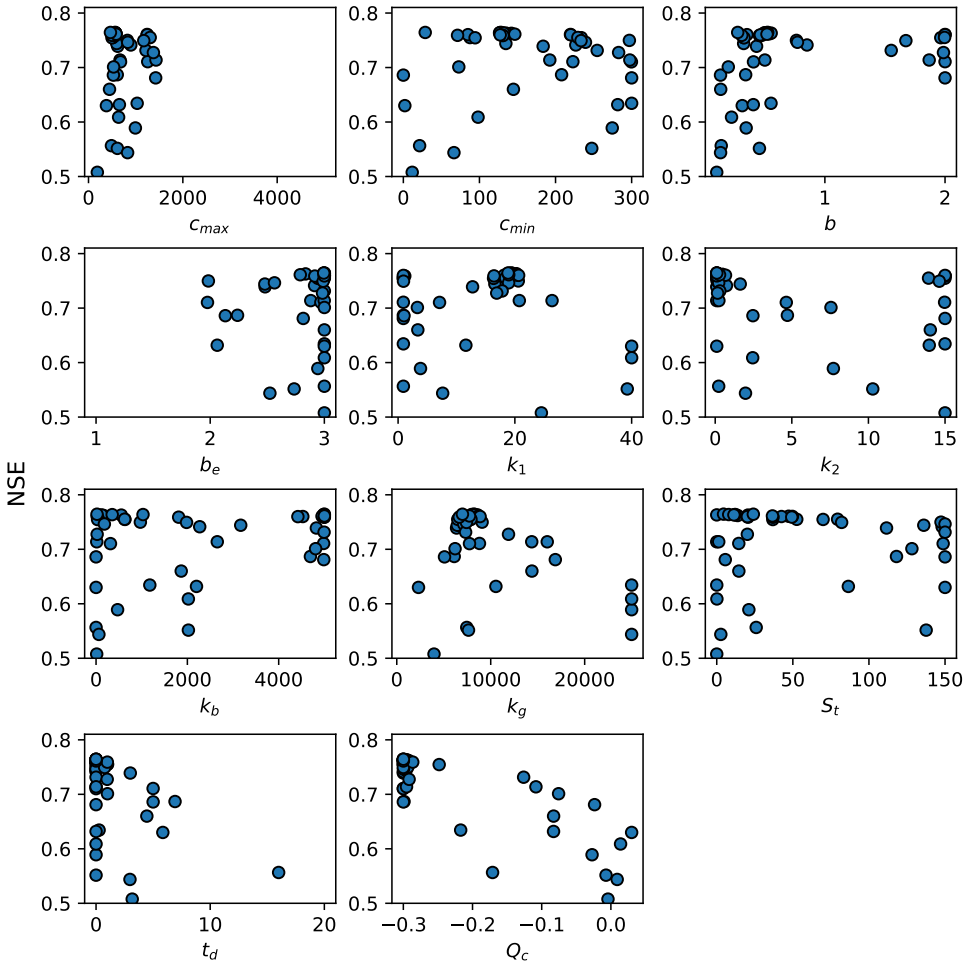

Figure 4.4: Dotty plots for the NM optimised parameters with NSE on the full period as metric.

overestimated. This underestimation of peak flow and consistent bias in very low flows is also observed for the PDM by Alvarez-Garreton et al. (2014)

4.3.3.2 Particle swarm optimisation

The resulting hydrograph of PSO optimisation is given in Figure 4.5 with $Q_{m,PSO}$ the modelled flow. The flow predictions largely overlap with those from NM calibration, so similar conclusions can be drawn for the most part. Performance metrics are again given in Table 4.1. Compared to the initial set, improvements are again seen for the calibration and full period while for the validation period both NSE and FHV deteriorate.

4.3.3.3 Comparison

Because differences between the NM and PSO hydrographs are hard to distinguish, an alternate representation of these flows is given in Figure 4.6. On the left (Fig-

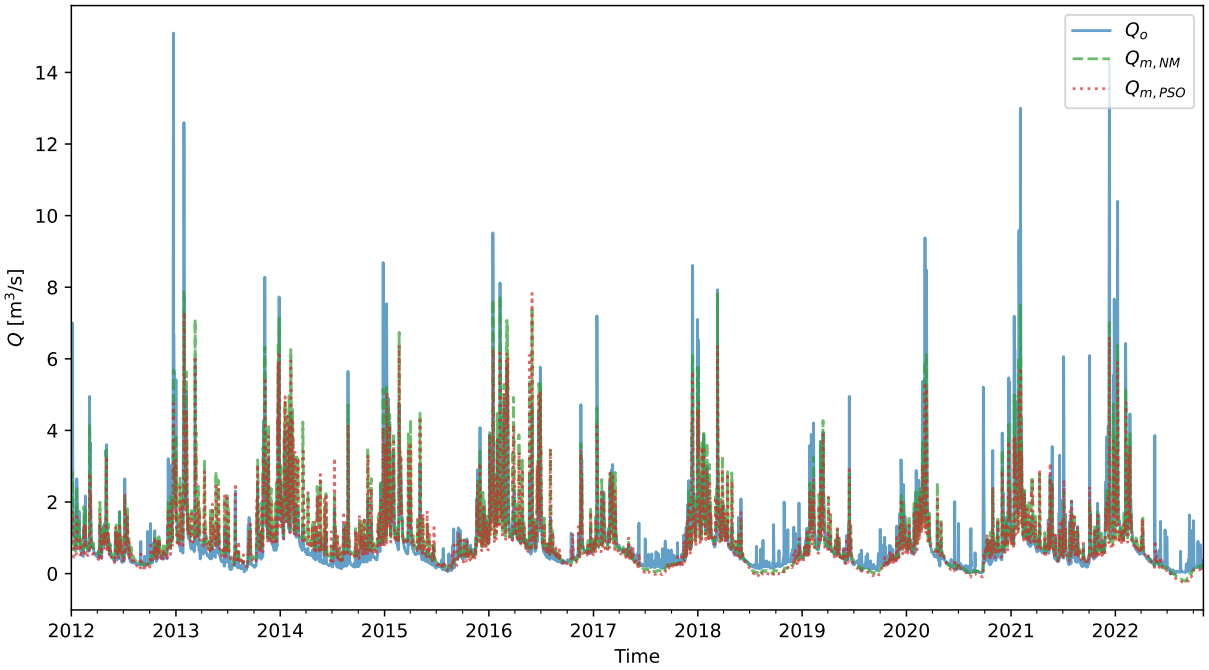

Figure 4.5: Hydrograph comparing observed Q_o with the modelled $Q_{m,NM}$ and $Q_{m,PSO}$.

ure 4.6 (a)) the flow-duration curve (FDC) is given, which is a plot of exceedance probability p on the horizontal axis and the ordered flow quantiles on the vertical axis (Vogel and Fennessey, 1994). The aforementioned overestimation of the initial parameter set is again notably present with a consistently higher p for the same Q compared to the observations. Both NM and PSO optimised parameters reduce this overestimation, with PSO performing best until $p \approx 0.75$, as $Q_{m,NM}$ shows some overestimation for mid-range values of p. For high values of p however, an underestimation occurs on very low baseflows which is worse for PSO than NM, as the former also produces negative flow estimates.

For assessing the performance on peak flow, the quantile-quantile (QQ) plot on Figure 4.6 (b) with observed and modelled ordered flow quantiles on the horizontal and vertical axis respectively, is more useful. For comparison with Figure 4.6 (a), a vertical line is drawn through the observed flow with $p = 0.02$ (cf. $p_{obs} = 0.02$ on Figure 4.6 (b)). Ideally, observed and modelled quantiles would be on the bisector (1:1). For NM and PSO, the quantiles are consistently below the bisector for peak flow indicating an underestimation. The latter is clearly worse for $Q_{m,PSO}$, as also reflected by the larger FHV for all three periods (cf. Table 4.1).

Lastly, the different C^* produced by the initial set (C^*_{init}), PSO ($C^*_{PSO,opt}$) and NM ($C^*_{NM,opt}$) are compared in Figure 4.7. Additionally, the 20 best performing parameters of the NM calibration are included to give an indication of variety in C^* modelling by displaying the mean ($\mu_{C^*_{NM}}$) and an interval around it from the 2.5$^{\text{th}}$ percentile ($C^*_{NM,2.5\%}$) to the 97.5$^{\text{th}}$ percentile ($C^*_{NM,97.5\%}$), calculated with the linear

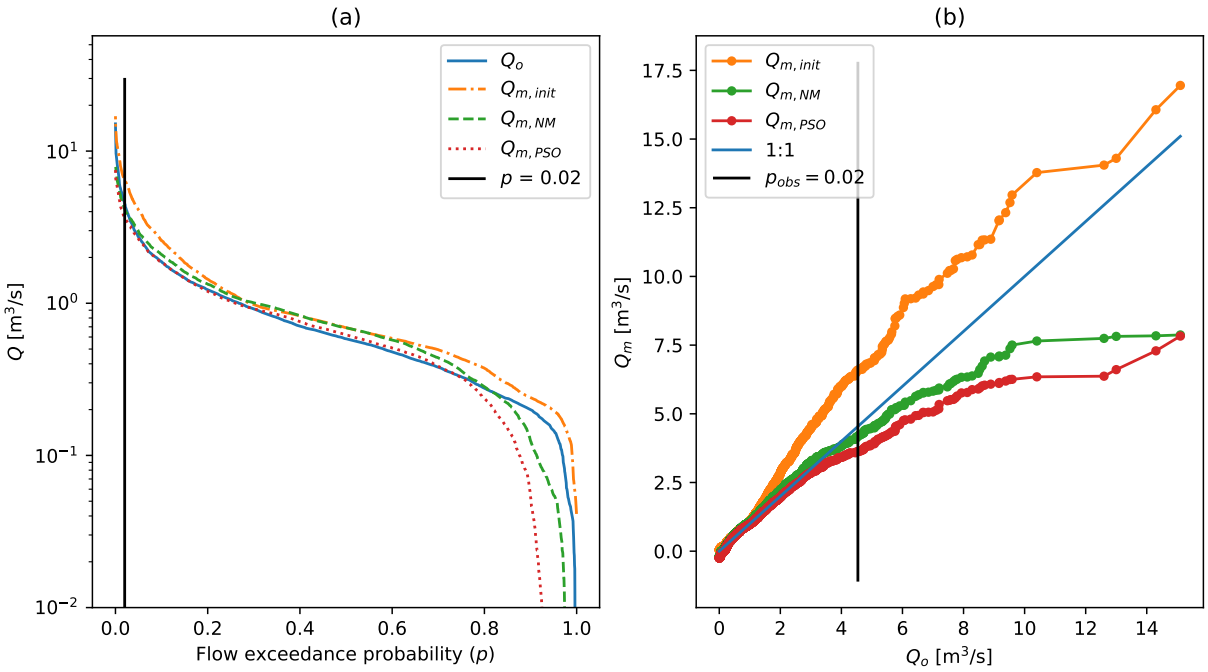

Figure 4.6: (a) FDC for observed Q_o compared to the three modelled scenarios: $Q_{m,init}$, $Q_{m,NM}$ and $Q_{m,PSO}$. (b) QQ plot with observed and modelled flow quantiles on the x-axis and y-axis respectively.

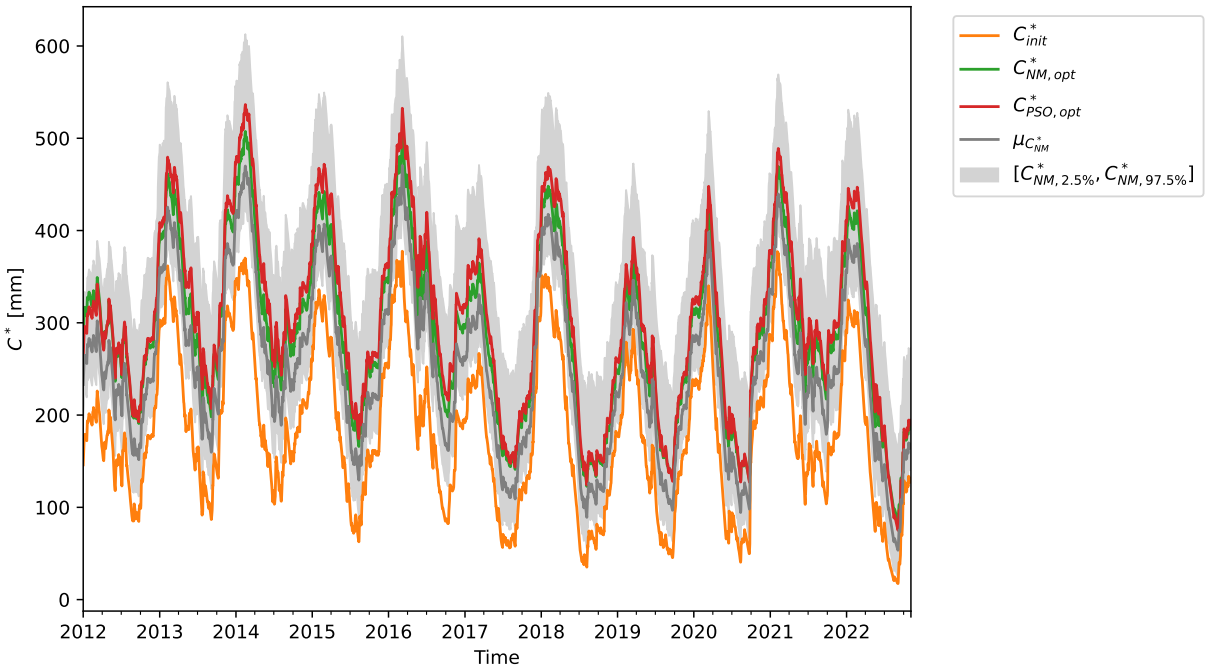

Figure 4.7: Time series of C^* with the initial (C^*_{init}), NM optimised ($C^*_{NM,opt}$) and PSO optimised ($C^*_{PSO,opt}$) parameter sets. Additionally, a time series of the average C^* for the 20 best performing NM optimised parameter sets ($\mu_{C^*_{NM}}$) is added combined withan interval from the 2.5th to the 97.5th percentile.

interpolation method (see Hyndman and Fan (1996)). Although clear differences are present in the absolute values of modelled C^*, they all follow a very similar

pattern and come very close to being vertical translations of one another. This is confirmed by Figure 4.8, which shows the Z-scores (i.e. the the variable minus its mean divided by its standard deviation) of the C^*_{init}, $C^*_{NM,opt}$, $C^*_{PSO,opt}$ and $\mu_{C^*_{NM}}$ time series. Only C^*_{init} shows a somewhat different pattern, while the Z-scores of the other three are almost identical. As the inverse observation operator in Chapter 5 will be trained on the Z-score, this could possibly allow the retrieved Z-score to be back transformed to the C^* of other parameter sets than the one initially used to calculate the Z-score.

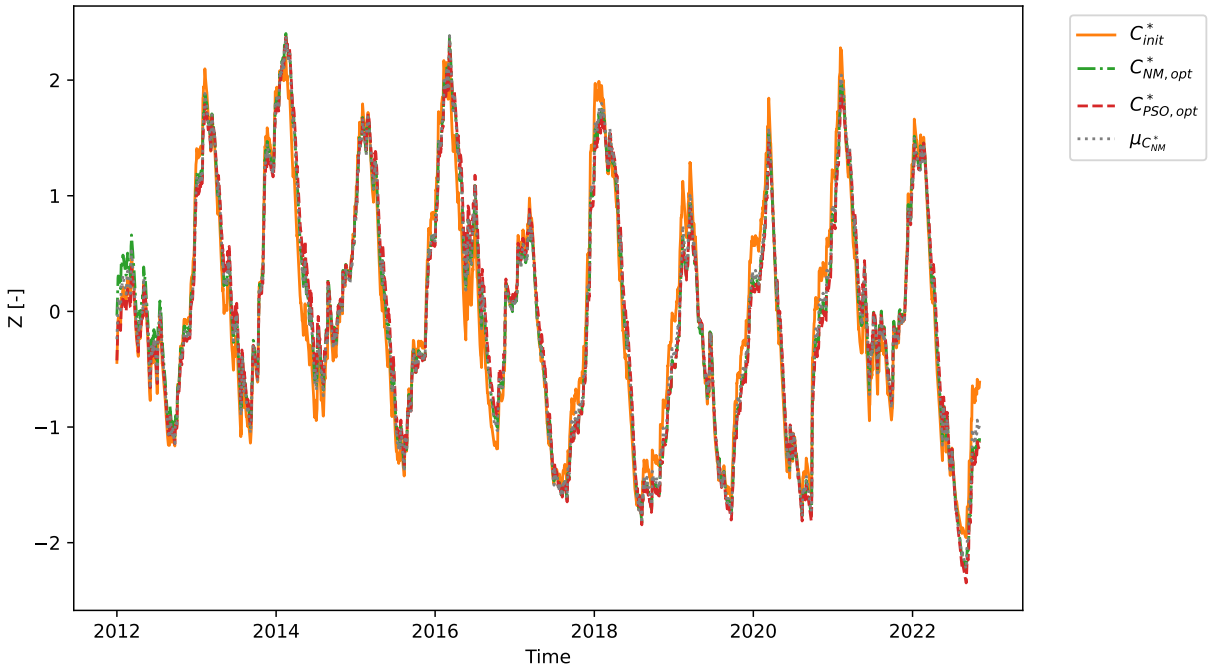

Figure 4.8: Z-score transformation of the four time series in Figure 4.7.

Based on the performance metrics of Table 4.1, it becomes apparent that the optimal NM parameter set (see Table A.1) performs best over most periods and for most metrics, as only the FHV on the validation period is significantly better for another parameter set. Therefore the NM parameter set will be used to provide C^* as estimation target in Chapter 5 and for modelling with DA in Chapter 6.

5. MACHINE LEARNING METHODS FOR THE INVERSE OBSERVATION OPERATOR

Vrij grondig gewijzigd en uitgebreid sinds ik dit laatste keer doorstuurde. Aangezien dit deel qua methode en resultaten misschien het belangrijkste deel is van de thesis, hoor ik hier (indien mogelijk natuurlijk) graag feedback van jullie beiden over.

With the observational time series processed to inputs (features) and C^* from the PDM as training targets, different machine learning (ML) algorithms can be evaluated to construct an empirical relationship between the two. After an introduction on the (inverse) observation operator in Section 5.1, the different ML methods will be explained in Section 5.2. Subsequently, the different features for the inverse observation operator are covered in Section 5.3, followed by the configuration and results for the different methods in Section 5.4 and 5.5 respectively.

5.1 The (inverse) observation operator for data assimilation

Although a more extended overview on the different methods for data assimilation (DA) will be given in Chapter 6, the key equation for DA and the concept of the observation operator are formalised here. As mentioned in Section 1.2, DA aims at optimally combining measurements with estimates of state variables produced by dynamic models (e.g. the PDM). With $\mathbf{x}$ the vector of state variables and $\mathbf{o}$ the vector of observations, DA can be summarised in a simplified manner as applying the update equation (Bouttier and Courtier, 2002; Reichle, 2008):

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k[\mathbf{o}_k - h_k(\hat{\mathbf{x}}_k^-)] \tag{5.1}$$

with the subscript k denoting that all of the parameters/variables/sates can be time-dependent (and are evaluated at the k^{th} time step for this example). The estimate of the states by the model $\hat{\mathbf{x}}_k^-$, called an priori estimate, is updated with the observational info $\mathbf{o}_k$ to $\hat{\mathbf{x}}_k^+$, the a posteriori state vector, which has the goal of being a better estimate of the true but unknown state of the system $\mathbf{x}_k$. To apply this updating via Equation 5.1, one needs to compare $\mathbf{o}_k$ with $\hat{\mathbf{x}}_k^-$. If the states of the model are observed, $\mathbf{o}_k$ can be written as $\mathbf{x}_{k,obs}$ and the term between square brackets in Equation 5.1 is replaced by $\mathbf{x}_{k,obs} - \hat{\mathbf{x}}_k^-$. The gain matrix $\mathbf{K}_k$ determines the strength

of the update, where the higher the value, the more the observation is trusted and influences $\hat{\mathbf{x}}_k^+$ (Reichle, 2008).

As explained in Section 1.2, states are rarely directly observable and hence observation operators, mapping $\hat{\mathbf{x}}_k^-$ to the space of $\mathbf{o}_k$, or retrieval algorithms/inverse observation operators, mapping $\mathbf{o}_k$ to $\mathbf{x}_k$, are necessary. When following the second option, Equation 5.1 can be reformulated as:

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k[h^{-1}(\mathbf{o}_k) - \hat{\mathbf{x}}_k^-] \tag{5.2}$$

with h^{-1} the inverse observation operator.

Since conceptual model states, in this case the C^* of the PDM, can not easily be physically related to measurable properties, empirical h^{-1} need to be used (cf. Section 1.2). For example in Aubert et al. (2003), in-situ measured volumetric SM is assimilated in the GR4J rainfall-runoff model. As SM is not a state variable of the GR4J model, a simple linear regression is used as observation operator with the measured SM as dependent variable and the water level in the conceptual SM reservoir divided by its maximum height as independent variable. For the PDM, DA of SSM obtained via RS, more specifically with a passive microwave radiometer, has been explored by Alvarez-Garreton et al. (2014). As SSM is not a state variable of the PDM, mapping of the observation to the S_1 state is done by linear regression or CDF matching (i.e. the SSM observation is transformed to the S_1 value with the same cumulative probability).

Because obtaining this SSM still requires a retrieval algorithm, a forward RTM called the land parameter retrieval model (Owe et al., 2008) for the SSM data of Alvarez-Garreton et al. (2014), it was proposed in Section 1.2 to replace this SSM retrieval algorithm combined subsequent empirical h^{-1} by a ML method. Of the aforementioned examples of this ML approach, the work of Rains et al. (2022) is most related to this dissertation. Rains et al. (2022) explore the use of support vector regression (SVR) as a forward observation operator mapping LAI and modelled SM to σ^0_{VV}. An inverted approach will be followed here to map the observations of γ^0 and LAI, combined with features related to the day-of-year (DOY) and orbit direction (cf. Section 5.3), to C^*. This ML-retrieved C^*_{obs} will then be used for DA in Chapter 6 with an assimilation scheme in the form of Equation 5.2.

5.2 Machine learning methods

Besides the classification of h^{-1} according to model structure, which will be used in what follows, the different inverse observation operators can also be classified according to the number of time steps used for producing one C^*_{obs} estimate. The first category only uses one time step per estimate and can be generally described

as:

$$C^*_{obs,k} = h^{-1}(\mathbf{o}_k) \tag{5.3}$$

with $\mathbf{o}_k$ the $(p \times 1)$ observational vector denoting the full set of p observed features at time step k. Alternatively, all observations within a certain time window can be used within the inverse observation operator model:

$$C^*_{obs,k} = h^{-1}(\mathbf{o}_k, \mathbf{o}_{k-1}, ..., \mathbf{o}_{k-\tau+1}) \tag{5.4}$$

with τ the number of time steps considered. In what follows, C^*_k (the target) and $C^*_{obs,k}$ (the h^{-1} estimate) will be replaced by y_k and $\hat{y}_k$ respectively for notational simplicity.

5.2.1 Linear methods

All linear models below have essentially the same model structure, given for time step k by:

$$\hat{y}_k = w_0 + \sum_{j=1}^{p} w_j o_{k,j} = w_0 + \mathbf{w}^T \mathbf{o}_k = \tilde{\mathbf{w}}^T \tilde{\mathbf{o}}_k \tag{5.5}$$

with w_0 the bias and w_1,...,w_p the remaining parameters. $\tilde{\mathbf{w}}$ is the full parameter vector $[w_0, w_1, \ldots, w_p]^T = [w_0, \mathbf{w}^T]^T$ and $\tilde{\mathbf{o}}_k = [1, \mathbf{o}_k^T]^T$ the modified vector of observations for time step k (Bishop, 2006a,b). The methods covered below differ in how the best suited value of $\tilde{\mathbf{w}}$ is obtained.

5.2.1.1 Linear regression

The most simple model considered is linear regression (LR) with multiple explanatory variables, which can be seen as an extension of the method used in Aubert et al. (2003). To obtain $\tilde{\mathbf{w}}$, the sum of squared residuals is taken as a loss function:

$$L(\tilde{\mathbf{w}}) = \sum_{k=1}^{N} (y_k - \tilde{\mathbf{w}}^T \tilde{\mathbf{o}}_k)^2 = (\mathbf{y} - \mathbf{O}\tilde{\mathbf{w}})^T (\mathbf{y} - \mathbf{O}\tilde{\mathbf{w}}) = ||\mathbf{y} - \mathbf{O}\tilde{\mathbf{w}}||_2^2 \tag{5.6}$$

with N the total number of time steps, $\mathbf{y}$ a $N \times 1$ vector of y_k values, $\mathbf{O} = \begin{bmatrix} \tilde{\mathbf{o}}_1^T \\ \tilde{\mathbf{o}}_2^T \\ \ldots \\ \tilde{\mathbf{o}}_N^T \end{bmatrix}$ an $N \times (p+1)$ matrix and $||\ldots||_2$ the Euclidean norm. $L(\tilde{\mathbf{w}})$ is effectively minimised by taking the gradient with respect to $\tilde{\mathbf{w}}$ and setting it to zero. This results in the ordinary least squares (OLS) estimate for $\tilde{\mathbf{w}}$:

$$\hat{\tilde{\mathbf{w}}} = (\mathbf{O}^T \mathbf{O})^{-1} \mathbf{O}^T \mathbf{y} \tag{5.7}$$

No iterative learning is required, as the above is a closed form solution (Bishop, 2006a; Hastie et al., 2009).

5.2.1.2 Ridge and lasso regression

To prevent overfitting when estimating the coefficients of the LR model, a regularisation term can be added to the loss function which penalises high values $\tilde{\mathbf{w}}$. A first option, called ridge regression (RR), penalises the quadratic Euclidean norm of $\tilde{\mathbf{w}}$:

$$L(\tilde{\mathbf{w}}) = ||\mathbf{y} - \mathbf{O}\tilde{\mathbf{w}}||_2^2 + \lambda ||\tilde{\mathbf{w}}||_2^2 \tag{5.8}$$

with λ the regularisation coefficient. λ is what's called a hyperparameter, the optimisation of which (for LR and other methods) is covered in Section 5.4.1. Following the same procedure as for LR, a closed form OLS estimate of $\tilde{\mathbf{w}}$ can be obtained:

$$\hat{\tilde{\mathbf{w}}} = (\lambda \mathbf{I} + \mathbf{O}^T\mathbf{O})^{-1}\mathbf{O}^T\mathbf{y} \tag{5.9}$$

Alternatively, one can also regularise $\tilde{\mathbf{w}}$ by penalising the absolute norm (which is the sum of the absolute values of the vector elements), in what is called lasso regression (LaR). Its loss function is:

$$L(\tilde{\mathbf{w}}) = ||\mathbf{y} - \mathbf{O}\tilde{\mathbf{w}}||_2^2 + \lambda ||\tilde{\mathbf{w}}||_1 \tag{5.10}$$

Interestingly, for large enough values of λ, some of the parameters in $\tilde{\mathbf{w}}$ can be driven to zero, effectively executing a feature selection yielding a more sparse model (Bishop, 2006a). Note that finding the minimum of Equation 5.10 is essentially a quadratic programming problem i.e. the minimisation of a multivariate quadratic function subject to linear constraints on the parameters (Hastie et al., 2009). In the utilised software (cf. Section 5.2.4), a slightly different form of Equation 5.10 is used by including a factor $\frac{1}{2n_{train}}$ (with n_{train} the number of training samples) before the squared loss (Pedregosa et al., 2011).

5.2.1.3 Support vector regression

Although support vector machines were initially developed for classification tasks, the structure was adapted to support regression with ϵ—support vector regression (SVR) (Smola and Schölkopf, 2004). Once more, a regularised loss function is constructed, but this time no loss value is assigned to predictions that lie within the so-called ϵ-insensitive tube, which is of width 2ϵ around the true y_k values. This is formalised as:

$$L(\mathbf{w}) = C\sum_{k=1}^{N} E_\epsilon(\hat{y}_k - y_k) + \frac{1}{2}||\mathbf{w}||_2^2 \tag{5.11}$$

with C the inverse regularisation parameter $\left(\sim \frac{1}{\lambda}\right)$ and $E_\epsilon(\hat{y}_k - y_k)$ the linear ϵ-insensitive loss function (Bishop, 2006b):

$$E_\epsilon(\hat{y}_k - y_k) = \begin{cases} 0 & \text{if } |\hat{y}_k - y_k| < \epsilon \\ |\hat{y}_k - y_k| - \epsilon & \text{otherwise} \end{cases} \tag{5.12}$$

where ϵ and C are the hyperparameters that require tuning.

5.2.2 Nonlinear methods

Only nonlinear methods which are not neural networks (NNs) are covered below, as NNs will be covered in section 5.2.3.

5.2.2.1 Nonlinear support vector regression

To extend linear methods to a nonlinear feature space, Equation 5.5 is reformulated as:

$$\hat{y}_k = \mathbf{w}^T \phi(\mathbf{o}_k) + w_0 = \tilde{\mathbf{w}}^T \phi(\tilde{\mathbf{o}}_k) \tag{5.13}$$

with $\phi(\mathbf{o}_k)$ a $(p' \times 1)$ vector with nonlinear so-called basis functions of the input vector $\mathbf{o}_k$. As a simple example, a possible polynomial expansion for a two-dimensional vector $\mathbf{o}_k = [o_{k,1}, o_{k,2}]^T$ is $\phi(\mathbf{o}_k) = [o_{k,1}^2, o_{k,2}^2, o_{k,1} o_{k,2}]^T$ (Bishop, 2006a).

In this context, the kernel function can be defined as:

$$k(\mathbf{o_k}, \mathbf{o'_k}) = \phi(\mathbf{o_k})^T \phi(\mathbf{o_k}) \tag{5.14}$$

This kernel can then be inserted into linear algorithms as defined by Equation 5.13 with the kernel trick. Although Equation 5.14 is defined as the explicit dot product of the transformed vectors, valid kernels also exist that are a direct function $\mathbf{o_k}$ and $\mathbf{o'_k}$ (Bishop, 2006c).

To extend ϵ-SVR to a nonlinear feature space, the loss function defined in Equation 5.11 has to be reformulated with the concepts of slack variables, which are a different way of penalising predictions outside of the ϵ-insensitive tube. This results in following conditions:

$$y_k \leq \hat{y}_k + \epsilon + \xi_k \tag{5.15}$$

$$y_k \geq \hat{y}_k - \epsilon - \hat{\xi}_k \tag{5.16}$$

which results in a modified constrained minimisation problem defined by:

$$\min_{\mathbf{w},\xi,\hat{\xi}} \left[C \sum_{k=1}^{N} (\xi_k + \hat{\xi}_k) + \frac{1}{2} ||\mathbf{w}||_2^2 \right] \tag{5.17}$$
$$\text{subject to } \xi \geq 0,\ \hat{\xi} \geq 0, \text{ Equation 5.15 and Equation 5.16}$$

Such problem can be solved by the introducing the Lagrangian:

$$\begin{aligned} \mathcal{L} = C \sum_{k=1}^{N} (\xi_k + \hat{\xi}_k) + \frac{1}{2} ||\mathbf{w}||_2^2 - \sum_{k=1}^{N} (\mu_k \xi_k + \hat{\mu}_k \hat{\xi}_k) \\ - \sum_{k=1}^{N} a_k (\epsilon + \xi_k + \hat{y}_k - y_k)) - \sum_{k=1}^{N} \hat{a}_k (\epsilon + \hat{\xi}_k - \hat{y}_k + y_k)) \end{aligned} \tag{5.18}$$

with $a_k \geq 0$, $\hat{a}_k \geq 0$, $\mu_k \geq 0$ and $\hat{\mu}_k \geq 0$ the Lagrange multipliers. The minimisation of Equation 5.17 can be simplified by considering that with the method of Lagrange multipliers, the gradients of $\mathcal{L}$ to $\mathbf{w}, b, \xi_k$ and $\hat{\xi}_k$ should all be zero (Bishop, 2006b). Substituting the results of above equalities in the Lagrangian results in a simplified optimisation problem that can be solved by software for quadratic programming, examples of which (together with a more extended mathematical elaboration on ϵ-SVR) are provided in Smola and Schölkopf (2004).

Lastly, it is interesting to remark that by substituting $\nabla_{\mathbf{w}} \mathcal{L} = 0$ into Equation 5.13 (for details, see Bishop (2006b)), an alternative formulation of the model is obtained:

$$\hat{y}(\mathbf{o}) = \sum_{k=1}^{N} (a_k - \hat{a}_k) k(\mathbf{o}, \mathbf{o}_k) + w_0 \tag{5.19}$$

which depends only on the kernel function of the features. Consequently, kernel functions can be used that are not easily expressed in terms of $\phi(\mathbf{o}_k)$ (Bishop, 2006b). In this dissertation, the Gaussian radial basis function (RBF) is used (Smola and Schölkopf, 2004; Han et al., 2012):

$$k(\mathbf{o}_k, \mathbf{o}'_k) = \exp\left(\frac{-||\mathbf{o}_k - \mathbf{o}'_k||_2^2}{2\sigma^2} \right) = \exp(-\gamma ||\mathbf{o}_k - \mathbf{o}'_k||_2^2) \tag{5.20}$$

This results in three hyperparameters: γ, ϵ and C

5.2.2.2 Gaussian process regression

Gaussian process regression (GPR) is generally defined as a distribution over functions:

$$f(\mathbf{o_k}) \sim GP(m(\mathbf{o_k}), k(\mathbf{o_k}, \mathbf{o_k}')) \tag{5.21}$$

where $f(\mathbf{o_k})$ denotes the actual (but unknown) function of $\mathbf{o}_k$, $m(\mathbf{o_k})$ the mean function and $k(\mathbf{o_k}, \mathbf{o_k}')$ the covariance function. To clarify above notation, it is in-

teresting to consider linear regression from a Bayesian perspective. For this, $f(\mathbf{o}_k)$ is given by Equation 5.13 (in its form without explicit bias w_0) and the prior normal distribution over $\tilde{\mathbf{w}}$ with mean $\mathbf{0}$ and covariance $\frac{1}{\alpha}\mathbf{I}$ is denoted as $N(\mathbf{0}, \frac{1}{\alpha}\mathbf{I})$. Rewriting Equation 5.13 for all N predictions gives:

$$\mathbf{y} = \mathbf{\Phi}\tilde{\mathbf{w}} \tag{5.22}$$

with $\mathbf{\Phi}$ being defined analogously to $\mathbf{O}$ in Equation 5.6. As $\mathbf{y}$ is just a linear combination of $\tilde{\mathbf{w}}$, it is also normally distributed with:

$$\begin{gathered} E[\mathbf{y}] = \mathbf{\Phi}E[\tilde{\mathbf{w}}] = \mathbf{0} \\ \text{cov}[\mathbf{y}] = E[\{\mathbf{y}) - E(\mathbf{y})\}\{\mathbf{y} - E(\mathbf{y})\}^T] = \mathbf{\Phi}E[\tilde{\mathbf{w}}\tilde{\mathbf{w}}^T]\mathbf{\Phi}^T = \frac{1}{\alpha}\mathbf{\Phi}\mathbf{\Phi}^T = \mathbf{K} \end{gathered} \tag{5.23}$$

where $\mathbf{K}$ is the so-called Gram matrix. The assumption of $m(\mathbf{x}) = 0$ is often used combined with the normalisation of $\mathbf{y}$.

Note that in reality, y_k is often only a noisy observation/realisation of the underlying true function, formalised as:

$$y_k = f(\mathbf{o}_k) + \epsilon \tag{5.24}$$

with ϵ following a Gaussian distribution with variance σ_n^2. With N training targets (combined in the $\mathbf{y}$ vector) and M test targets (combined in $\mathbf{f}_*$), the prior situation is defined by:

- a training Gram matrix $\mathbf{K}$ of size $N \times N$ with the element on row i and column j being $\mathbf{K}_{i,j} = k(\mathbf{o}_i, \mathbf{o}_j)$
- a $M \times M$ test Gram matrix $\mathbf{K}_{**}$ with $\mathbf{K}_{**i,j} = k(\mathbf{o}_{*i}, \mathbf{o}_{*j})$ with $\mathbf{o}_{*i}$ and $\mathbf{o}_{*j}$ being general notations for the features of a test point

The joint distribution of $\mathbf{y}$ and $\mathbf{f}_*$ is the multivariate Gaussian defined by:

$$p(\mathbf{y}, \mathbf{f}_*) = N\begin{pmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_* \end{pmatrix}, \begin{matrix} \mathbf{K} + \sigma_n^2\mathbf{I} & \mathbf{K}_* \\ \mathbf{K}_*^T & \mathbf{K}_{**} \end{matrix}\Bigg) \tag{5.25}$$

with $\boldsymbol{\mu} = \boldsymbol{\mu}_* = 0$ and and $\mathbf{K}_{*i,j} = k(\mathbf{o}_i, \mathbf{o}_{*j})$ with $\mathbf{K}_*$ of size $N \times M$. Note that $\sigma_n^2\mathbf{I}$ is added to the training covariance matrix assuming independent noise. The key in GPR is now to go from the joint distribution to the conditional distribution over $\mathbf{f}_*$ on the features and training targets:

$$\begin{gathered} p(\mathbf{f}_*|\mathbf{y}, \mathbf{O}, \mathbf{O}_*) = N(\mathbf{K}_*^T(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y}, \mathbf{K}_{**} - \mathbf{K}_*^T(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{K}_*) = \\ N(\boldsymbol{\mu}_{*,posterior}, \mathbf{K}_{**,posterior}) \end{gathered} \tag{5.26}$$

$\boldsymbol{\mu}_{*,posterior}$ is used as the prediction with a 95% confidence interval (CI) defined by $[\boldsymbol{\mu}_{*,posterior} - 2\text{tr}(\boldsymbol{K}_{**,posterior}), \boldsymbol{\mu}_{*,posterior} + 2\text{tr}(\boldsymbol{K}_{**,posterior})]$ (Bishop, 2006c; Rasmussen and Williams, 2006).

Analogous to ϵ-SVR, the RBF kernel is used. Again hyperparameters are present: the length scale σ of the kernel and the noise variance σ_n^2. For a more in-depth introduction to GPR, the reader is referred to Rasmussen and Williams (2006).

5.2.3 Neural networks

5.2.3.1 Multilayer perceptron

The core unit of the multilayer perceptron (MLP) (and more generally any NN) is the neuron/perceptron. The latter is defined as a nonlinear transformation of the basic linear model structure of Equation 5.5 (with $o_{k,j}$ generalised to x_j):

$$a = g\Big(\sum_{j=1}^{p} w_j x_j + w_0\Big) \tag{5.27}$$

with $g()$ the nonlinear activation function and a the activation. It is the combination of multiple of these neurons in series and parallel that results in the MLP structure. This is visually summarised in Figure 5.1 (Razavi, 2021). Note that neuron is used for continuous activation functions, while the perceptron refers to the original discrete activation function of Rosenblatt, making neuron the preferred term (Bishop, 2006d).

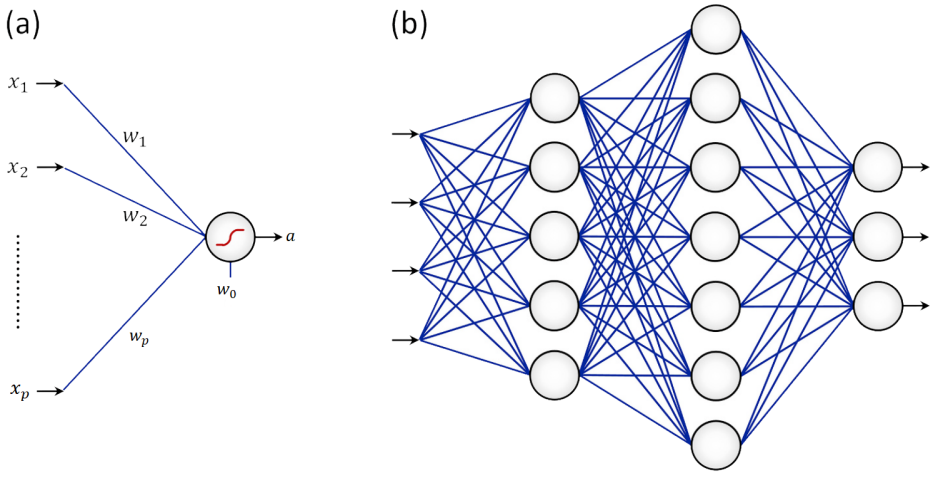

Figure 5.1: (a) A single neuron/perceptron (b) a MLP with four inputs, two hidden layers and three outputs, adapted from Razavi (2021).

Key hyperparameters are the depth (i.e. the number of layers) of the network and the width (i.e. the number of neurons) per layer. Also the activation function could be tuned (by for example comparing the hyperbolic tangent, sigmoid...), but here the rectified linear unit (ReLU) activation ($g(z) = max(0, z)$) is used, as it is a good default choice (Goodfellow et al., 2016a). On Figure 5.1, one can differentiate between the first two layers with activation functions, called hidden layers, and the

final output layer, for which no activation is used in regression (as this allows outputs in the range $]-\infty, \infty[$) (Ranjan, 2020a; Razavi, 2021).

The key challenge for using the MLP is finding suitable values for the parameters and biases (from here on jointly denoted as $\mathbf{w}$) (Razavi, 2021). For this purpose, one evaluates the gradient of the loss function $L(\mathbf{w})$ with regard to $\mathbf{w}$: $\nabla_{\mathbf{w}} L(\mathbf{w})$. Here, the mean squared error (MSE) will be used as $L(\mathbf{w})$. To evaluate $\nabla_{\mathbf{w}} L(\mathbf{w})$ in an efficient way, backpropagation is used, which in essence is a recursive application of the chain rule (Goodfellow et al., 2016a). The simplest way to now update $\mathbf{w}$ is with gradient descent:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \epsilon \nabla_{\mathbf{w}} L(\mathbf{w}_t) \tag{5.28}$$

with t the iteration step and ϵ the learning rate (set to 10^{-3}). The gradient will not be calculated for all training points at once but only for a subset, the so-called minibatch (Goodfellow et al., 2016b). In what follows, the more advanced Adam algorithm will be used. Adam is still a first-order optimisation method, but it aims at estimating curvature (i.e. second order) information by calculating estimates of the first and second order moments of the gradient (Kingma and Ba, 2014). In this context, an epoch is defined as one complete iteration of the full dataset (in multiple minibatches) (Ranjan, 2020a).

Due to the many degrees of freedom related to the large number of parameters, it is important to avoid overfitting and lack of generalisation. Therefore, a part (20%) of the training data will be withheld as validation data to assess generalisation performance. Two regularisation approaches will be used (Goodfellow et al., 2016c):

1. Early stopping: The parameter set and number of epochs is returned for the lowest validation error (instead of from the last epoch). Then, the NN is retrained on the entire training data with this recommended number of epochs ($n_{epochs,rec}$).
2. Dropout: For each minibatch, an input and/or hidden unit is exclued with an assigned probability ($p_{dropuout}$) during training. In this way, it is prevented that certain neurons stop participating in prediction (Ranjan, 2020a). In this dissertation, only dropout after hidden layers is considered.

While the biases are initialised at zero, the weights are initialised by sampling from $U\left(-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\right)$ with n the width from the previous layer (Glorot and Bengio, 2010). To account for this stochasticity and because of the local optimisation, it is recommend to start the optimisation from different starting sets of initial weights (Bishop, 2006d). Analogous to the work of Kratzert et al. (2019b) on long short-term memory (LSTM) networks, each model will therefore be initialised eight times and trained for 100 epochs while checking the generalisation performance on the validation data. The average of the $n_{epochs,rec}$ (≤ 100) from these eights initialisations with valida-

tion data, is then used as the number of epochs for training on the full training dataset, which is again performed from eight initialisations.

5.2.3.2 Long short-term memory

For problems with inputs related in time, MLPs are limited in the sense that they only provide a static mapping between one input and one output. By incorporating signals from previous time steps, a network structure called the recurrent neural network (RNN) is obtained. The long short-term memory (LSTM) is a type of RNN with a model structure better adapted at learning long-term dependencies by addressing the issue of vanishing/exploding gradients in training (Staudemeyer and Morris, 2019).

The idea of looping within the NN structure is displayed in Figure 5.2, with the memory cell (also called recurrent cell) equivalent to the neural network, $\mathbf{x}_i$ ($i \in \{1, \ldots, t\}$) the inputs (with subscripts denoting the time steps), $\mathbf{h}_i$ the hidden states and y_i the outputs. So at one time step, where $\mathbf{x}_i$ is used as input for the NN, it calculates both an output y_i and a hidden state $\mathbf{h}_i$ that is passed on to the next time step, when the NN has $\mathbf{x}_{i+1}$ as input. This concept is clarified by the so-called unrolling in time of the RNN as depicted in Figure 5.2 (Olah, 2015). Because of the resemblance between the unrolled structure and the MLP, backpropagation can be applied on the former for training in what is called backpropagation through time (Staudemeyer and Morris, 2019).

The structure of one recurrent cell, is given in Figure 5.3 (with (t) equivalent to $_i$ above). This illustration clarifies that in between times steps, two different states are passed on: the cell state $\mathbf{c}(t)$ (not to be confused with c from the PDM, cf. Section 4.2), which can be regarded as the long-term memory, and the hidden state (and output) $\mathbf{h}(t)$ related to short-term memory. By passing the input vector of the current time step $\mathbf{x}(t)$ combined with the states from the previous time step $\mathbf{h}(t-1)$ and $\mathbf{c}(t-1)$ through several gates (i.e. forget, input and output), which are essentially just neurons (with sigmoid (σ) or tanh as activation functions) combined with matrix calculations, temporal dynamics can be predicted (Ranjan, 2020b).

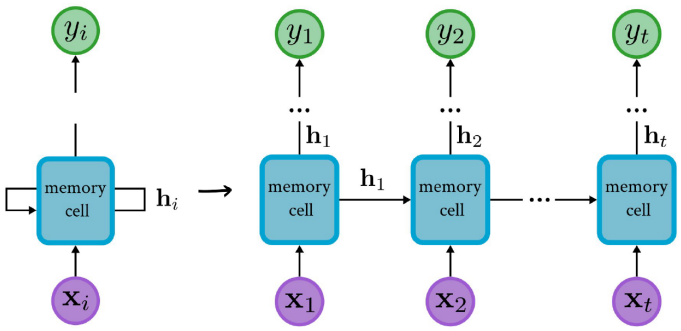

Figure 5.2: Unrolling the RNN in time, adapted from Olah (2015).

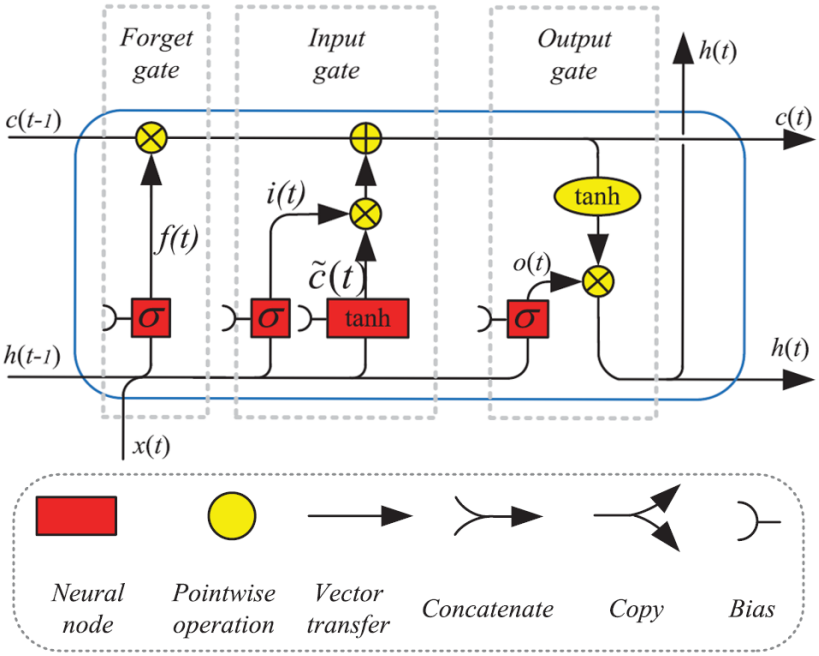

Figure 5.3: The recurrent cell for the long short-term memory (LSTM), adapted from Yu et al. (2019).

In this dissertation, a many-to-one LSTM variant will be used, for which only the hidden state at the time step of prediction, $\mathbf{h}_t$, is used to generate a scalar output y_t (the C^*_{obs}) by applying a linear combination on the elements of $\mathbf{h}_t$ in a so-called linear dense layer. Note the equivalency here with Equation 5.4: $\mathbf{o}_k = \mathbf{x}_t$ and $\mathbf{o_{k-\tau+1}} = \mathbf{x}_1$. In other words, τ inputs are passed through the LSTM for each prediction.

For training and regularisation, the same methods will be followed as described for the MLP, with dropout only being applied to the inputs of the linear dense layer. The difference lies in the hyperparameters for the LSTM, which are the number of inputs in the time window (τ) and the dimensionality (n_{hidden}) of the **h** and **c** vectors.

5.2.4 Software for the machine learning methods

For all the linear and nonlinear methods which are not NN, the `scikit-learn` package from Pedregosa et al. (2011) is used. For the NNs, `TensorFlow` (Abadi et al., 2015) and `Keras` (Chollet et al., 2015) are used. For all methods covered above, it is infeasible to describe the implementation specifics in all detail. Therefore, the reader is referred to the documentation of above packages for more details and can assume that (unless otherwise specified) the default choices are respected.

5.3 Feature engineering and exploration

To incorporate information on orbit direction and DOY, which are categorical features, following two methods are applied:

1. For orbit direction, one column is created called ascending that is one if the orbit is ascending and zero otherwise (i.e. descending). This is done as an alternative for the bias correction for each relative orbit as applied Rains et al. (2022).
2. The DOY is translated to a continuous, numeric signal by constructing both a sine and cosine wave with a period of one year: $\sin\left(\frac{\text{DOY}}{365.2425/(2\pi)}\right)$ and $\cos\left(\frac{\text{DOY}}{365.2425/(2\pi)}\right)$ (TensorFlow, 2022).

With these added features, a total of 12 (= p) explanatory variables are obtained jointly denoted as $\mathbf{o}_k$ (cf. Section 5.2): three VV-related, three VH-related, three LAI-related, two DOY-related and one orbit-related feature(s). Note that all features and targets are standardised to zero mean and unit variance to limit the effect of variables with different scales and have more robust optimisation in training.

Analogous to the calibration and validation period of Chapter 4, a training and test period need to be defined for the ML training:

1. Training period: 10/06/2015 until 31/12/2020
2. Test period: 01/01/2021 until 03/11/2022

5.3.1 Features for time window methods

RNN such as the LSTM are designed to deal with regular time intervals, which is not the case for the current dataset. Many options exist to deal with irregular intervals such as imputation, interpolation, adapted RNN structures etc. (see Weerakody et al. (2021)), but the simplest is adding Δt, the number of days between current and previous observation, as a feature to obtain the so-called RNN-Δt (Rubanova et al., 2019). The irregularity of Δt is displayed on Figure 5.4 and is mostly present in the beginning of the observations, certainly in the period before the launch of S-1-B in mid 2016, and after the premature end of the S-1-B mission in 2022 (cf. Section 3.2.1). The average Δt is 3.6 days.

For the non-NN time window methods, the input at time step k is simply a concatenation of the inputs as given in Equation 5.4: $[\mathbf{o}^T_{k-\tau+1}, \ldots, \mathbf{o}^T_{k-1}, \mathbf{o}^T_k]$. For the LSTM, these τ inputs are not concatenated, but passed through the recurrent cell one after the other: first $\mathbf{o}_{k-\tau+1}$, then $\mathbf{o}_{k-\tau+2}$... and finally $\mathbf{o}_k$. This means that contrary to the other time window methods, the number of model parameters does not increase

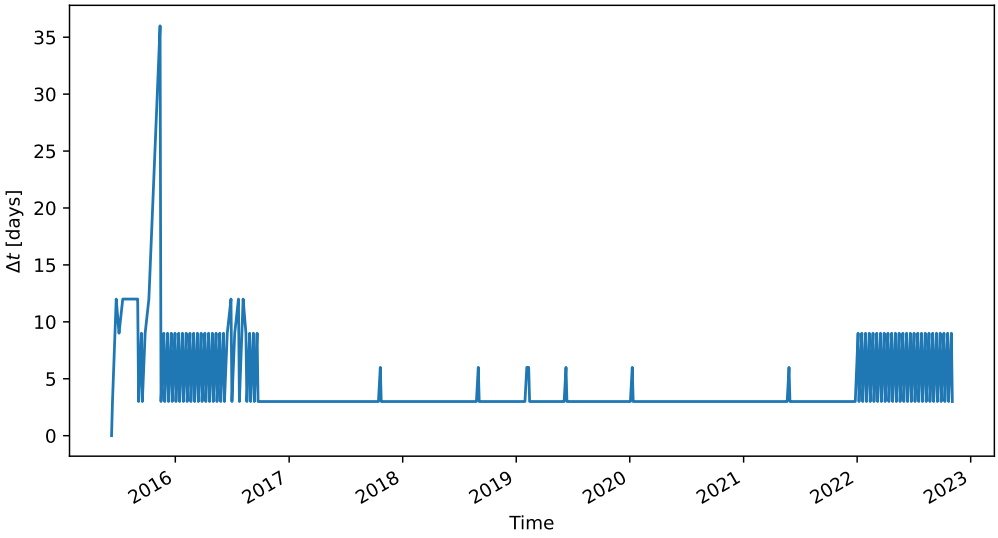

Figure 5.4: Number of days between consecutive S-1 observations.

with increased τ for the LSTM. The obvious disadvantage of all the time window methods is that prediction can only start after τ observations have been made.

5.3.2 Features and target correlations

Before training on the features as described above, the Pearson correlation coefficient (ρ) between them is analysed (see Figure 5.5). Most striking is the large correlation between the different γ^0 features and between the different LAI features, despite averaging over different land uses. Also VV and VH γ^0 time series show large correlations.

In classical statistical modelling, one wants to use regression models for e.g. testing the statistical significance of an explanatory variable. For this purpose, it is crucial to avoid collinearity i.e. a large correlation between inputs, and avoiding inputs with $\rho > 0.7$ is therefore generally advised. In the field of ML however, one is generally less interested in interpretation of the coefficients. Furthermore, ML techniques are more robust to collinearity if regularisation is applied (Dormann et al., 2013), as is the case for many of the methods presented above. Nonetheless, a reduced dataset which averages out leaf area index (LAI), γ^0_{VV} and γ^0_{VH} over the pasture and agriculture (excluding forest cf. Section 3.3.2) at once is constructed. Its correlations are shown in Figure 5.5 (b), indicating a reduction in collinearity, but no elimination as high correlation ($\rho > 0.7$) is still present between γ^0_{VV} and γ^0_{VH}.

On Figure 5.6, the Pearson correlation between the explanatory variables and C^* is shown, split up between ascending (C^*_{asc}) and descending (C^*_{desc}) orbits. It is apparent that for both the full (a) and reduced (b) dataset, the ascending orbit shows better correlations between γ^0 and C^*. For the full dataset, the highest correlation

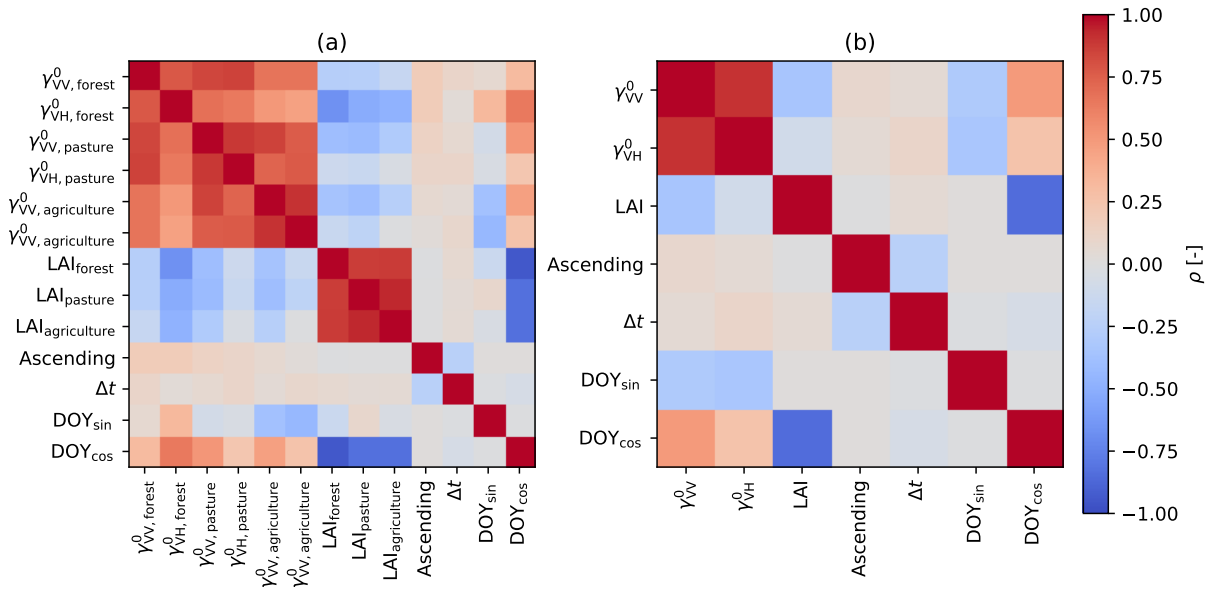

Figure 5.5: Pearson correlation (ρ) between input features for the full dataset (a) and the reduced dataset averaged out over the tree land use at once (b).

($\rho_{max} = 0.7060$) between C^* and γ^0 is found for averaging over forest land use. Initially, this is counter-intuitive as based on SAR physics (cf. Section 3.3.2), SM should be the least correlated to forest backscatter. This could be explained by looking at the LAI, which is negatively correlated to the cosinusoidal signal (implying a clear seasonality, cf. Section 2.4.2) and γ^0 (cf. Figure 5.5). The above indicates that when LAI is low (in the winter), both C^* and γ^0 are high and vice versa in summer. This seasonal change in backscatter, most prominent for VH, is also observed by Dostálová et al. (2018) for deciduous forests and attributed to the seasonal change in tree foliage, which can cause more attenuation of the signal or forward scattering away from the sensor. A potential additional source of higher backscatter in winter are the openings created in the canopy, allowing double bounce scattering to occur on the stems (Rosenqvist and Killough, 2018). In summary, the correlation between γ^0_{forest} and C^* is related to an analogous seasonality caused by different underlying phenomena, signalling that caution is needed when including this feature. Note that DOY_{sin} also shows high correlation with C^* because of the inherent seasonal trend of the former, but this is not considered problematic as the goal of including this feature is providing a consistent indicator of seasonality.

5.4 Experimental design: hyperparameters and model inputs

Besides the model structures themselves, two large degrees of freedom remain: the hyperparameters and the choice of input variables.

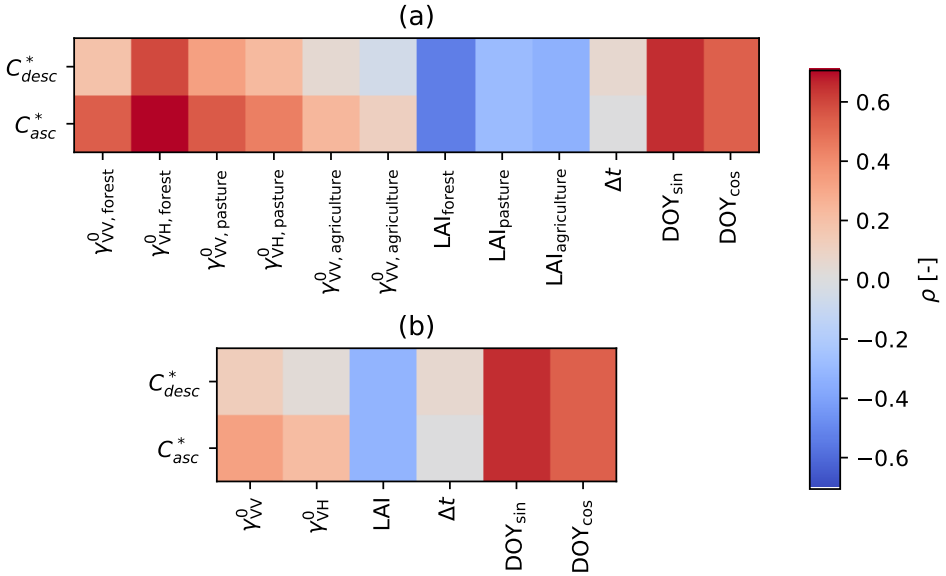

Figure 5.6: Pearson correlation between C^*, split up between ascending and descending, and the explanatory variables for the full (a) and reduced (b) dataset.

5.4.1 Hyperparameter optimisation

For (nearly) all non-NN methods, hyperparameter optimisation is done by k-fold cross-validation (CV). When multiple hyperparameters are optimised, each over a certain range, all combinations are considered and the name grid search is often assigned to this method (Smets et al., 2007). In this approach, the training data is divided into k (approximately) equally sized groups called folds. Note that the test data cannot be used for optimising the hyperparameter(s), as these are used for assessing the model performance one the optimal hyperparameter(s) is/are obtained. For the first iteration of k-fold CV, the first fold is used as validation set over which the score is calculated, while the other $k-1$ folds are used for training the model. This is repeated until every fold has been the validation set, after which the mean (and, if of interest, the standard deviation) over the validation scores can be calculated. So for selecting an optimal (combination of) hyperparameter(s), k—fold CV is applied for all these hyperparameter combinations, and the combination with the best average score is deeded optimal, as it has shown the best generalisation on unseen data (James et al., 2021). As the data considered is a time series, it is important to keep the order as is when constructing folds to prevent an overfitted model (Bergmeir and Benítez, 2012). Here, k is set to five and the coefficient of determination (R^2) will be used as score:

$$R^2(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{\sum_{k=1}^{N}(y_k - \hat{y_k})^2}{\sum_{k=1}^{N}(y_k - \bar{y})^2} \tag{5.29}$$

Note that the GPRs are an exception, as their hyperparameters are trained by maximising the likelihood of the hyperparameters given the training observations

(Bishop, 2006c). Within `scikit-learn`, this maximisation is repeated from different initial values of the hyperparameters by using the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm, which is a quasi-Newton method.

For hyperparameter optimisation of NNs, generalisation performance will only be assessed on one validation dataset (i.e. the 20% holdout on the training data) instead of performing the CV. For the reasons as described in Section 5.2.3, the training will be started from multiple initialisations, but for computational reasons this number is reduced from eight to four.

Table 5.1: The data-driven models and their hyperparameters. Time window method is abbreviated as TW. A number of $\log_{10}$ equally spaced values between min and max is abbreviated as logrange(min,max,number). If sources are used for deriving the hyperparameter range, they are given after the model name.

Model	**Hyper-paramter**	**Range**	**Default/ Initial**
Linear regression	-	-	-
Linear regression TW	τ	-	5
Ridge regression	λ	logrange(10^{-3},10^{3},100)	-
Lasso regression	λ	logrange(10^{-3},10^{3},100)	-
Ridge regression TW	λ	logrange(10^{-3},10^{3},100)	-
	τ	[1, 2, 3, 4, 5, 10, 20, 30, 40, 50, 60]	5
Lasso regression TW	λ	logrange(10^{-3},10^{3},100)	-
	τ	[1, 2, 3, 4, 5, 10, 20, 30, 40, 50, 60]	5
Linear ϵ-SVR (Smets et al., 2007)	C	logrange(10^{-10},10^{3},14)	1
	ϵ	logrange(10^{-3},10^{1},20)	0.1
RBF ϵ-SVR (Smets et al., 2007; Rains et al., 2022)	C	log(10^{-10},10^{3},14)	1
	ϵ	-	0.1
	γ	log(10^{-5},10^{5},50)	-
GPR	σ	$\log(\sigma_1), \ldots, \log(\sigma_{100}) \sim U(-2, 2)$	-
	σ_n	$\log(\sigma_{n,1}), \ldots, \log(\sigma_{n,100}) \sim U(-1, 3)$	-
MLP (Ranjan, 2020a; Lievens, 2023)	Layers	[2,3]	2
	Neurons per layer	[4,12,20,28]	$p/2$ and $p/4$ or $2p$ and p
	$p_{dropout}$	[0,0.2,0.4]	0 or 0.2
LSTM (Kratzert et al., 2018, 2019b)	n_{hidden}	[4,8,12,16]	10
	τ	[30,60,90]	100
	$p_{dropout}$	[0,0.2,0.4]	0

In Table 5.1, the hyperparameters for the different methods are covered. For GPR the interpretation is again slightly different: the range of hyperparameters given contains only the starting points for the optimisations (cf. supra), not the final values. If only one hyperparameter is tried or certain hyperparameters are tried before performing hyperparameter optimisation, it is put in the default/initial column. For

the MLP, each of the layers can differ in the number of neurons. As initial structure, $p/2$ and $p/4$ neurons for the first and second layer respectively are tried according to the rule of thumb by Ranjan (2020a), stating that nodes in a layer should be approximately one half of the number of inputs. A second initial structure based on Lievens (2023), suggest taking number of nodes in the first layer to be $2p$, after which the halving rule of Ranjan (2020a) is again applied for the second layer. For LSTM, the initial τ is set to 100 to approximate one year sequence input length as used in Kratzert et al. (2018), while hyperparameter optimisation is inspired by Kratzert et al. (2019b).

5.4.2 Input selection

Besides trying both the full and reduced datasets as defined in 5.3.2 as inputs, also within these datasets specific selections of inputs can be made. More concretely, it will be investigated for the full dataset what the merit is of including forest-related features for the aforementioned reasons of physical implausibility (cf. Section 5.3.2). For both the full and reduced data set, the effect of excluding the DOY-related features is examined to investigate how important this feature is to the correct prediction of the model. A heavy dependence on it would not be a desired characteristic, as the goal is to extract information from the observational γ^0 and LAI, with the DOY ideally only being auxiliary.

5.5 Results

5.5.1 Linear methods

5.5.1.1 Linear regression

Performance scores for LR over the various input combinations are given in Table 5.2. Out of the single time step inputs, LR performs best on the full dataset. Excluding DOY- or forest-related features both lead to a drop in performance, albeit more pronounced for DOY. For the reduced dataset, the significant drop in performance when excluding DOY indicates that the seasonality of the sine/cosine wave is leveraged heavily for prediction, which is not desired.

For the time window method with $\tau = 5$ (on the full dataset), the training performance increases but at the cost of lower test performance, indicating overfitting and a lack of generalisation. For LR, one time step inputs are therefore preferred

Combined with the $C*$ targets as provided by the PDM, the retrieved C^*_{obs} for training and testing are visualised for four different scenarios in Figure 5.7: full dataset

(a), full dataset without DOY (b), without forests (c) or for a time window input (d). Going from (a) to (b), removing the smooth seasonal effect of the sinusoidal inputs clearly results in a more jagged retrieved C^*_{obs}. The changes from (a) to (c) are less pronounced, but most noticeable is that (c) is less capable of retrieving the very high (winter 2021) and very low (summer 2022) C^* values. Lastly (d) produces the smoothest time series out of the depicted scenarios, but at the cost or larger deviations in the test period.

Table 5.2: Comparison of R^2 for the various input combinations with LR. Best results per period are displayed in bold. "-" denotes no alterations to the dataset as described above.

Period	**Full**				**Reduced**	
	-	No DOY	No forest	Time window	-	No DOY
Train	0.8220	0.7591	0.7956	**0.9034**	0.7781	0.1102
Test	**0.8082**	0.7405	0.7806	0.7695	0.7165	0.0382

5.5.1.2 Lasso and ridge regression

Training LaR and RR on the full dataset with $\tau = 1$, results in worse performance than the unregularised LR (cf. Table 5.3 for LaR/RR). This behaviour could be expected, as LR already generalises well for $\tau = 1$, indicating that regularisation might not be necessary. Note that the difference in order of magnitude for λ for RR and LaR can be attributed to the different formulations of the loss functions (cf. Section 5.2.1.2).

As performed for LR, both LaR and RR are also trained on the full dataset with $\tau = 5$. More regularisation occurs compared to $\tau = 1$ for RR, but (counter-intuitively) less for LaR, as reflected by the respective rise and drop in λ (cf. Table 5.3). In both cases however, the models generalise better than time window LR as illustrated by the higher R^2_{test}, with LaR showing the superior performance of the two. Interestingly, the feature selection provided by LaR (cf. Section 5.2.1.2) sets the weights of 16 of the 60 ($=\tau \cdot p = 5 \cdot 12$) inputs to zero, which are mostly (10/16) related to DOY.

Table 5.3: Performances of LaR and RR for different input configurations. Best resulst per period are displayed in bold.

Dataset	**Model**	$\mathbf{R^2_{train}}$	$\mathbf{R^2_{test}}$	$\boldsymbol{\lambda}$	$\boldsymbol{\tau}$
Full	Ridge	0.8052	0.7519	30.54	1
		0.8703	0.8138	70.55	5
	Lasso	0.8173	0.8027	$5.33\ 10^{-3}$	1
		0.8937	**0.8291**	$2.32\ 10^{-3}$	5
CV: full, no forest	Ridge	0.9691	0.6514	46.42	30
	Lasso	0.9512	0.6365	$1.32\ 10^{-3}$	20
CV adapted: full, no time	Ridge	0.9632	0.8127	46.41	30
	Lasso	**0.9729**	0.8255	$1.74\ 10^{-3}$	30

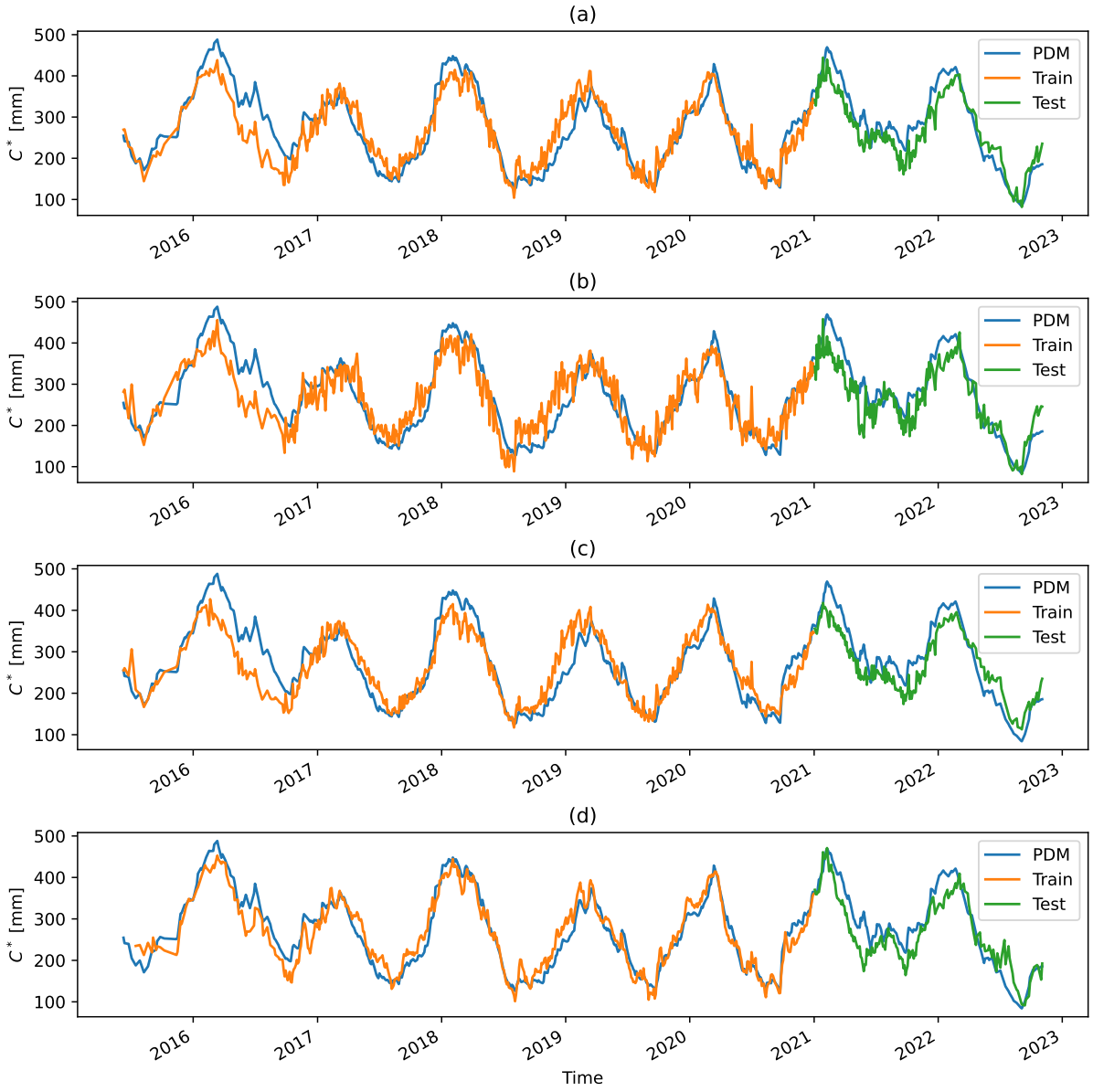

Figure 5.7: C^*_{obs} predictions by the LR model: (a) full dataset (b) full dataset without DOY (c) full dataset without forest (d) full dataset with time window of $\tau = 5$. The targets, as provided by the PDM, are displayed in blue.

When applying grid search over all the hyperparameter (λ and τ, cf. Table 5.1) and input (cf. Section 5.4.2) options, the feature set with best CV score is the full set without forest features. The optimal τ for RR and LaR are 30 and 20 respectively, with the former outperforming the latter with regards to CV and training score. Despite the CV procedure, both models are severely overfitted with R^2 scores being approximately 0.30 lower for testing than for training.

Because training LaR on $\tau = 5$ indicated that a lot of the information regarding DOY is superfluous, it is also investigated how the, according to the CV, best hyperparameters and inputs with no DOY perform. This does seem to somewhat alleviate the problem of overfitting, with the R^2 difference between training and testing drop-

ping to approximately 0.15. Nevertheless, none of the CV-optimised structures can outperform LaR with $\tau = 5$ trained on the full data for the test period. For reference, the configurations with the best R^2_{test} and R^2_{train} respectively (cf. Table 5.3) are also plotted in Figure 5.8: LaR with $\tau = 5$ on full dataset (a) and with $\tau = 30$ on full dataset without DOY (b). While (a) is quite analogous to the LR case (cf. Figure 5.7 (d)), albeit performing better on the test period, (b) on the other hand differs substantially from above predictions by showing very little short-term fluctuations.

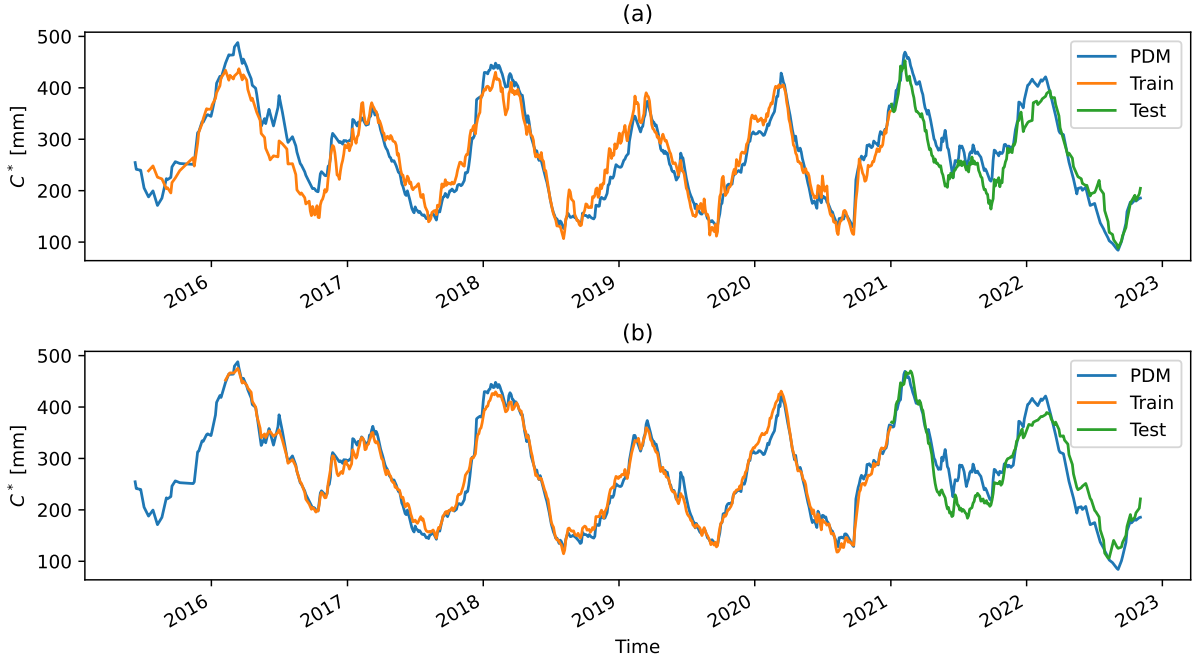

Figure 5.8: C^*_{obs} predictions by the LaR model: (a) full dataset with $\tau = 5$ (b) full dataset without DOY for $\tau = 30$.

5.5.1.3 Linear support vector regression

For linear ϵ-SVR, only one time step inputs on the full dataset are considered, as input alterations again show the same trends as for LR. In Table 5.4, two (linear) experiments are displayed: one with the default hyperparameters and one with the CV-optimised ones. For the default hyperparameters, the best R^2_{test} thus far is obtained. Although the CV procedure produces a slightly higher R^2_{train}, this is at the cost of an 0.03 drop in R^2_{test}. Consequently, the default hyperparameter set is preferred.

Table 5.4: Performances for both linear and nonlinear ϵ-SVR on full dataset. Best performances per period are displayed in bold.

Kernel	Configuration	R^2_{train}	R^2_{test}	C	ϵ	γ
Linear	Default	0.8139	**0.8311**	1	0.1	-
	CV-optimised	0.8159	0.8010	0.1	0.3360	-
RBF	Default	0.9058	0.8045	1	0.1	$1.73\ 10^{-3}$
	CV-optimised	**0.9442**	0.8058	10	0.1	$4.72\ 10^{-2}$

5.5.2 Nonlinear methods

5.5.2.1 Nonlinear support vector regression

As for the linear ϵ-SVR, default and CV-optimised hyperparameters for the full dataset are compared. The default γ is determined as $1/n_{train}$ (Pedregosa et al., 2011). Note that contrary to the linear case, ϵ is not optimised and set to 0.1 as done by Rains et al. (2022). Optimising ϵ was attempted but yielded worse generalisation and is hence not further covered.

By introducing the nonlinearity with the RBF kernel, higher training scores ($R^2 > 0.9$) can be achieved than with linear single time step methods. However just as for the time window methods, this elevated training score does not translate to superior generalisation on the test data. Therefore, it is not further investigated how nonlinear methods (such as ϵ-SVR) perform with time window inputs, as this would most likely aggravate the overfitting.

5.5.2.2 Gaussian process regression

As preliminary experiments indicated the large sensitivity of GPR towards the hyperparameters for prediction, only the optimised hyperparameters are discussed, with the performances presented in Table 5.5. For the full dataset, the optimised GPR yields a very slight improvement over RBF ϵ-SVR. Due to the similarity in performance between both methods, GPR is deemed representative to assess the added value of these nonlinear methods compared to simple LR for alterations of the input dataset.

Excluding DOY or forest features from the full dataset, results in very limited change of R^2_{train}, but R^{test}_2 drops significantly, even below the values obtained for LR (cf. Table 5.2). This also occurs when training on the reduced dataset with DOY features. So clearly, overfitting occurs in all the situations described above. Only for the reduced dataset without DOY, an improvement is obtained over LR, that is nonetheless still subpar compared to the predictions on more feature-rich datasets. Furthermore, it is apparent that the estimate of the noise variance (σ_n) via hyperparameter optimisation is higher for the reduced datasets. This increasing σ_n is

paired with lower length scales (σ), indicating the predictions are less influenced by trainig points farther away in the feature space.

In summary, the GPR are not capable of better C^*_{obs} retrievals than LR over the test period for less feature-rich datasets. For reference, the predictions of the best performing single time step nonlinear method, i.e. GPR on the full dataset, is given in Figure 5.9 with its 95% CI (cf. Section 5.2.2.2). Compared with the highest scoring R^2_{train} for the time window method (see Figure 5.8 (b)), more short term variation is present in the retrievals.

Table 5.5: Performances of GPR for different input configurations. Best results per period are displayed in bold.

Dataset	R^2_{train}	R^2_{test}	σ_n	σ
Full	**0.9448**	**0.8128**	0.1	2.59
Full: no DOY	0.9423	0.7002	0.1	1.85
Full: no forest	0.9340	0.7617	0.1	2
Reduced	0.8882	0.6725	0.133	1.59
Reduced: no DOY	0.3199	0.1830	0.73	1.18

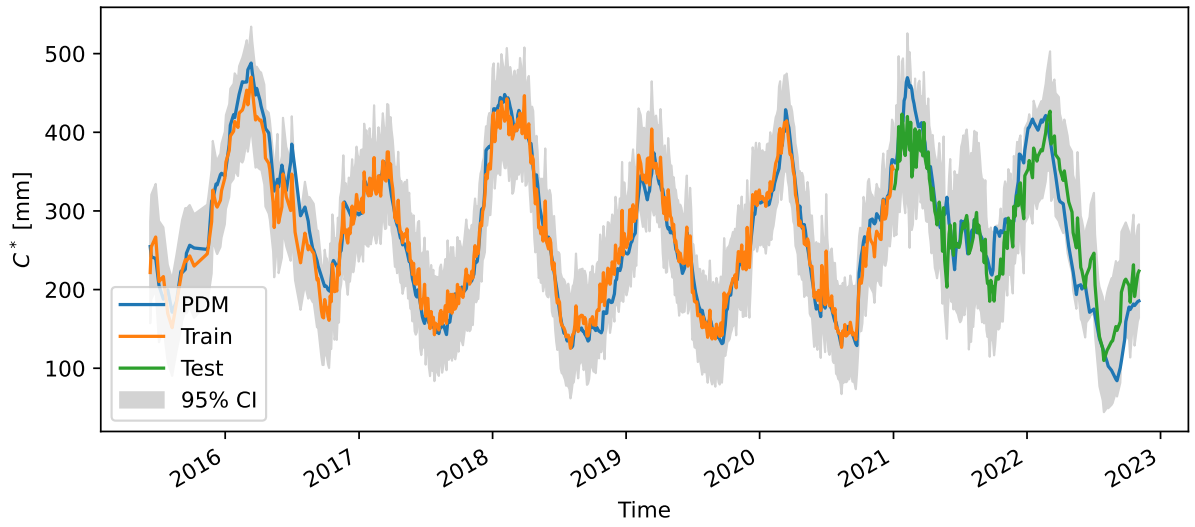

Figure 5.9: C^*_{obs} predictions and CI by GPR on the full dataset.

5.5.3 Neural network methods

Just as for GPR, also the MLP will be evaluated on its performance on both full and reduced datasets. For the time window method, the LSTM will be considered as an advanced alternative to the LaR and RR covered above.

5.5.3.1 Multilayer perceptron

In Table 5.6, the performances of several experiments with the MLP structure are covered. Not one score, but an average, minimum and maximum R^2 out of the eight initialisations of weights are given (cf. Section 5.2.3.1). Results on the reduced dataset are omitted since the same trend as for the other methods occurs: a drop in performance. Out of the initial configurations (defined according to the heuristics in Section 5.4.1), the larger network shows the best performance, albeit still worse than the non-NN nonlinear methods (and even worse than the linear methods with regards to the test period). The small configuration is the worst performer on the full dataset thus far, with greatly varying R^2 depending on the initialisations of the weights. Grid search over the hyperparameters and inputs (only reduced vs. full dataset considered), does not yield better performances. Reasons as to why the MLP might fall short, are discussed in Section 5.5.4

Table 5.6: Performances of the MLP on the full dataset. For R^2, the mean is given combined with (minimum, maximum). The number of nodes are ordered per layer. Best performances per period are in bold.

Configuration	$\mathbf{R^2_{train}}$	$\mathbf{R^2_{test}}$	**layers**	**nodes**	$\mathbf{p_{dropout}}$	$\mathbf{n_{epochs,rec}}$
Initial (Ranjan, 2020a)	0.7756 (0.6569, 0.8040)	0.6552 (0.3087, 0.7885)	2	6,3	0	89
Initial* (Lievens, 2023)	**0.8256** (0.7983, 0.8446)	**0.7475** (0.6836, 0.7890)	2	24,12	0.2	61
CV	0.7449 (0.6364, 0.7969)	0.5929 (0.3657, 0.6736)	2	28,4	0	18

5.5.3.2 Long short-term memory

As the LSTM is a time window method, the effect of whether or not to include DOY is tested on the initial hyperparameters, as for LaR/RR excluding it benefited generalisation. For LSTM, the opposite seems to be true, as reflected by the drop in R^2_{test} once excluded (cf. Table 5.7, again showing average, minimum and maximum R^2). Once more, the hyperparameter and input optimisation (considering all the input possibilities of Section 5.4.2) does not yield better generalisation. When compared to the linear time window methods (cf. Section 5.5.1.2), similar R^2_{train} are obtained but no LSTM configuration reaches the test performance of the optimal LaR (which has $R^2_{test} > 0.8$). The latter is confirmed on Figure 5.10, indicating how all eights models are overfitted in the training regime, while diverging from the targets for the test period.

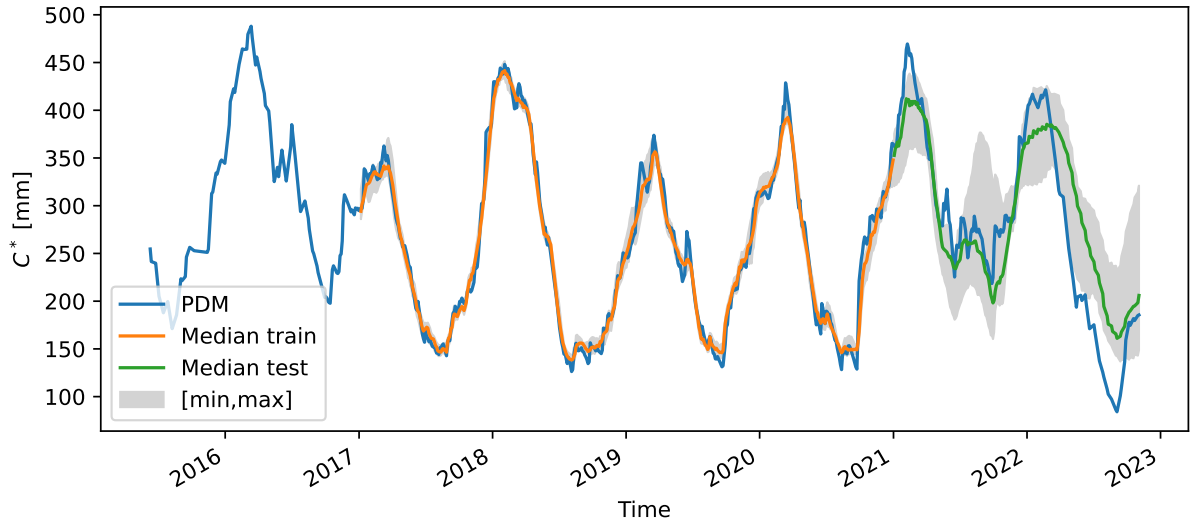

Figure 5.10: C^*_{obs} predictions and by LSTM on the full dataset. The grey interval goes from minimum to maximum predictions out of the eight initialised models.

Table 5.7: Performances of the LSTM. For R^2, the mean is given combined with (minimum, maximum). Best performances per period are in bold.

Dataset	$\mathbf{R^2_{train}}$	$\mathbf{R^2_{test}}$	$\mathbf{n_{hidden}}$	$\boldsymbol{\tau}$	$\mathbf{p_{dropout}}$	$\mathbf{n_{epochs,rec}}$
Full	0.9470 (0.9236, 0.9694)	**0.7521** (0.6598, 0.8513)	10	100	0	37
Full: no DOY	**0.9796** (0.9695, 0.9884)	0.7309 (0.6393, 0.8003)	10	100	0	71
CV: Full, no DOY	0.9588 (0.9483, 0.9672)	0.6700 (0.6164, 0.8006)	12	30	0.4	46

5.5.4 Comparison and discussion of methods

An interesting observation for a number of methods, is that despite extensively trying many combinations with a CV (or validation) procedure, this did not always result in better generalisation. This could be related to testing too many hypotheses, which can result in overfitting on the (cross-)validation data i.e. "by chance" a combination of hyperparmeters/inputs is found that gives a high (cross-)validation score (Ng, 1997).

For the neural networks, the main hypothesis for explaining the subpar performance is the lack of data. Despite starting from the large, high-resolution RS dataset, the spatial averaging produces a tabular dataset of (maximally) 13 features and around 600 training points. Consequently, the larger NNs (e.g. the LSTM) have a number of parameters equal to or lager than the number of training points. For these small datasets, it is not uncommon for NNs to perform worse than more

classic ML techniques. To circumvent this issue, solutions like pre-training could be explored (Feng et al., 2019). Furthermore, also training of the networks with second order methods, such as the quasi-Newton L-BFGS, could be tried, as for smaller networks and datasets, storing and calculating (the approximation of) the Hessian is feasible (Goodfellow et al., 2016b). Initial experiments with `scikit-learn`, which has an implementation of the L-BFGS optimisation for MLP, show that this could be a promising approach.

When comparing the models, the one time step linear models show the best generalisation, with R^2_{test} being generally only slightly lower than R^2_{train} and both > 0.8 for the best performers. For the nonlinear non-NN single time step methods, higher $R^2_{train}(> 0.9)$ were obtained, but no meaningful improvements occur on R^2_{test}, indicating overfitting by introducing the nonlinearity. This is analogous for the time window methods, with both linear (LaR/RR) and nonlinear (LSTM) methods showing overfitting, albeit worse for LSTM.

For DA in Chapter 6, not all of the models covered above will be tested. When selecting a subset of models, it is important to consider that the highest score in retrieval, does not necessarily translate to the best performance in DA. For example in Rains et al. (2022), the use of SVR as observation operator yielded retrievals more correlated to the observations than a semi-empirical approach, but nonetheless the latter performed better in DA. With this perspective in mind, following models will be analysed for DA:

1. LR full: LR on the full data as a baseline model with good generalisation.
2. LR full no forest: LR on the full data with no forest features for the reasons described in Section 5.3.2.
3. LaR full $\tau = 30$: LaR with $\tau = 30$ on full dataset without DOY to test the effect of assimilating these very smoothed out C^*_{obs} predictions.
4. Linear ϵ-SVR full: Linear $\epsilon-$SVR on the full dataset with default parameters as this models shows the highest R^2_{test}.
5. GPR full: GPR on the full dataset as the best performing nonlinear single time step model.

6. DATA ASSIMILATION

6.1 Overview of methods for data assimilation

Volledig nieuw stuk. Indien mogelijk, opnieuw graag feedback van jullie beiden hier over.

With the introduction on the observation operator covered in Section 5.1, the following focuses on the different mathematical methodologies for data assimilation (DA). Two large classes of methods can be considered: sequential and variational (Ide et al., 1997; Houser et al., 2010). To understand how the two are connected, it is easiest to start from a simplified problem with a scalar state variable. The goal is to obtain a best estimate of the true but unknown state x given the available information from both the model and observations. Using the notation from Section 5.1 whilst dropping the time subscript k for notational simplicity, this results in the minimisation of following objective function:

$$J = \frac{(x - \hat{x}^{-})^2}{\sigma_m^2} + \frac{(x - o)^2}{\sigma_o^2} \tag{6.1}$$

with σ_m^2 and σ_o^2 the model and observation error variance respectively. In variational DA, extended versions of J are minimised directly with numerical methods. This is often applied in numerical weather prediction, as the considerably sized state and observations vectors result in σ_m and σ_o being very large covariance error matrices, rendering an analytical solution of the minimisation infeasible to calculate (Reichle, 2008).

In sequential methods on the other hand, the error covariance matrices are explicitly considered. For the simple example, setting the derivative of Equation 6.1 to zero allows for solving to the form of the update equation (cf. Equation 5.1) with the gain equal to $\sigma_m^2/(\sigma_m^2 + \sigma_o^2)$ (Reichle, 2008). Sequential DA is used (amongst others) in hydrological forecasting (Plaza Guingla et al., 2013) and land surface modelling (De Lannoy et al., 2022; Houser et al., 2010). The fundamental difference between the several sequential methods lies in the construction of the gain matrix $\mathbf{K}$. The simplest DA method is direct insertion, where $\hat{\mathbf{x}}^{+} = \mathbf{o}$ (assuming an observed state). Consequently, model/observation uncertainties are not taken into account and the assumption is made that the observations are right. Slightly more advanced is Newtonian nudging, first described by Stauffer and Seaman (1990), which takes into account the quality of the observations and their distribution in space and time. For linear dynamic models, the Kalman filter provides the best

linear unbiased estimate for $\hat{\mathbf{x}}^+$ assuming known normally distributed errors. The Kalman filter has been extended to nonlinear systems with the extended Kalman filter, which relies on a linearisation of the dynamic model, and the ensemble Kalman filter, which uses an ensemble of models instead of linearisation (Houser et al., 2010). Most advanced is the use of the particle filter, which drops the assumption of the Gaussian distributions of errors (Plaza Guingla et al., 2013).

6.2 Newtonian nudging

In this dissertation, only a first exploratory attempt at DA is undertaken. Therefore, the relatively simple scheme of Newtonian nudging is chosen over the more advanced Kalman-based or particle filers. Furthermore, it is also used by Brocca et al. (2010) for the assimilation of a RS-derived SSM-related product in a conceptual rainfall-runoff model.

In the original paper of Stauffer and Seaman (1990), two versions of the algorithm are given, depending on how the a priori modelled state is relaxed towards the observed state: using an interpolation of the observations to the model grid or using individual observations. Here, the latter option is used. Although normally presented as a nudging term added to the dynamical equation describing the state variable (see for example Houser et al. (1998)), here the algorithm is given in the form related to the update equation, adapted from Martens et al. (2016) to the notation of Section 5.1 (without time subscript):

$$\hat{x}^+ = \hat{x}^- + G\frac{\sum_{i=1}^{N} W_i^2(x_i, y_i, z_i, t_i)\gamma_i\left(o_i - \hat{x}^-\right)}{\sum_{i=1}^{N} W_i^2(x_i, y_i, z_i, t_i)} \tag{6.2}$$

with N the number of observations used for updating, G the nudging factor, γ_i the quality factor of observation i, W_i an interpolation weight and o_i for simplicity considered a direct observation of the state. If a distributed model is used, W_i is a function of the observation's location (x_i, y_i, z_i in 3D) and time (t_i). As the PDM is a lumped model structure, only temporal weighing is considered. Replacing x with the state variable of interest C^* and considering that only one observation is assimilated at time, Equation 6.2 simplifies to:

$$C^{*+} = C^{*-} + G\gamma W_t(t)\left(C^*_{obs} - C^{*-}\right) \tag{6.3}$$

with $W_t(t)$ the temporal weighing function. The latter is trapezoidal and given by (Paniconi et al., 2003):

$$W_t(t) = \begin{cases} 1 & \text{if } |t - t_o| < \tau_a/2 \\ \frac{\tau_a - |t-t_o|}{\tau_a/2} & \text{if } \tau_a/2 \leq |t - t_o| \leq \tau_a \\ 0 & \text{if } |t - t_o| > \tau_a \end{cases} \tag{6.4}$$

with t_o the time of observation, t the time of the dynamic model and $\tau_a = t_a/2$ where t_a is the time window of assimilation. For τ_a, several values will be tried: 1,2,5,12,24 and 36 hours. The simplification is made that the Belgian standard time (UTC+01:00) is equal to the local solar time and hence, 18:00 and 6:00 for ascending and descending respectively (cf. Section 3.2.1) can be used as hours of passage.

If the errors on the model and observation (σ_m and σ_o) are known, the optimal γ is calculated as (Martens et al., 2016):

$$\gamma = \frac{\sigma_m}{\sigma_m + \sigma_o} \tag{6.5}$$

Obtaining these errors is not straightforward however. σ_m could be determined dynamically in time by setting up a model ensemble. For σ_o, a relationship dependent on a dynamically varying vegetation index exist in the case of SM (Martens et al., 2016). C^* on the other hand does not have an obvious expression for σ_o. Although one could argue that the uncertainty on the retrievals given by the inverse observation operators (e.g. from GPR) could be used as σ_o, this is not deemed fit as it would reflect the uncertainty of the inverse observation operator model and not of the factors (such as vegetation) affecting the retrieval accuracy. Combined with the added complexity of setting up an ensemble for σ_m, the choice is made to experiment with several static γ values instead (0.1, 0.25, 0.5, 0.75), where a higher γ reflects less trust in the modelled C^{*-}. G is set to 1 for a maximum effect of assimilation

6.3 Methodology

To compare discharge predictions by applying DA with those without DA, called open loop (OL), four different periods in time are evaluated, as presented in Table 6.1. The goal of P1 through P3 is to assess DA for different combinations of PDM and ML calibration/training, while the full period serves as global evaluation. The three evaluation metrics of Chapter 4 will again be used: NSE, mNSE and FHV. For these metrics, Δ is used to denote the difference between DA and OL performance. For FHV, the difference between the absolute values is taken, Δ|FHV|, as in this way Δ|FHV| < 0 is an improvement irrespective of the sign of FHV. As a starting point, the assimilation will be performed for the five inverse observation operators outlined in Section 5.5.4 with $\tau_a = 5$ h and $\gamma = 0.5$, displaying equal trust in the

model and the observations (cf. Section 6.4.1). Subsequently, DA with the different γ and τ_a values (as outlined in Section 6.2) will be assessed in Section 6.4.2.

Table 6.1: The different periods for assessment of DA performance.

Period	Start	End	PDM calibrated	h^{-1} trained
P1	01/06/2015	31/12/2019	✓	✓
P2	01/01/2020	31/12/2021	✗	✓
P3	01/01/2021	05/11/2022	✗	✗
PFull	01/06/2015	05/11/2022	-	-

6.4 Results

6.4.1 Initial assimilations

In Table 6.2, the differences in NSE between DA and OL are displayed. Most apparent is that for all periods and h^{-1} options, the impact of assimilation is limited with $|\Delta\text{NSE}| < 0.003$. Furthermore, it is clear that the highest R^2 for retrieval ($R^2_{\text{retrieval}}$ as displayed in Table 6.2) does not necessarily entail the largest improvement in NSE. For example excluding the forest feature from the dataset results in a lower $R^2_{\text{retrieval}}$ for LR, but yields a marginally higher ΔNSE. This non-straightforward relationship between $R^2_{\text{retrieval}}$ and ΔNSE, is most apparent for GPR and time window LaR, as these methods have the highest $R^2_{\text{retrieval}}$ (>0.9) on the training period (P1 + P2) but their ΔNSE is lower than for the other methods, certainly for P1. This is not illogical however, as when $R^2_{\text{retrieval}}$ approaches one, the difference between C^* and C^*_{obs} also approaches zero, resulting in no assimilation impact and a ΔNSE of 0. This point is also made by Rains et al. (2022), who therefore warn for the risk of overfitting when training a data-driven (inverse) observation operator.

Table 6.2: ΔNSE between DA and OL for the different periods and observation operators. The largest improvement per period is in bold. For reference, R^2 on the retrievals from Chapter 5 are given.

Inverse observation operator	ΔNSE				$R^2_{\text{retrieval}}$	
	P1	P2	P3	PFull	P1 + P2	P3
LR full	$1.31\ 10^{-3}$	$2.59\ 10^{-3}$	$4.37\ 10^{-4}$	$1.14\ 10^{-3}$	0.8220	0.8082
LR full no forest	$1.46\ 10^{-3}$	$2.34\ 10^{-3}$	$6.58\ 10^{-4}$	$\mathbf{1.25\ 10^{-3}}$	0.7956	0.7806
LaR full $\tau = 30$	$1.66\ 10^{-4}$	$2.09\ 10^{-3}$	$\mathbf{1.44\ 10^{-3}}$	$1.01\ 10^{-3}$	0.9729	0.8255
Linear ϵ-SVR full	$\mathbf{1.62\ 10^{-3}}$	$\mathbf{2.85\ 10^{-3}}$	$1.93\ 10^{-4}$	$1.21\ 10^{-3}$	0.8159	0.8311
GPR full	$5.72\ 10^{-4}$	$1.81\ 10^{-3}$	$8.31\ 10^{-5}$	$5.60\ 10^{-4}$	0.9448	0.8128

Although the differences are minimal, the linear methods outperform the nonlinear and time window one in most periods. Only for P3, time window LaR yields the largest improvement. The results for the other metrics, mNSE and FHV, are given

in Table A.2 and A.3 respectively. For the mNSE, similar trends as for the NSE are present, with the same h^{-1} models performing best for the same periods. The only difference is that ΔmNSE is slightly negative for P3 with linear ϵ-SVR and GPR. For Δ|FHV|, only very minor improvements ($\lesssim 0.5\%$) are obtained in P1,P2 and P3. Over PFull, the bias even increases slightly (< 0.15 %). With improvements on all three subperiods, this seems contradictory at fist. However, the increase is related to FHV being calculated on the top 2% highest discharges of a given time series (Q_m^h and Q_o^h cf. Section 4.3). Consequently, the shorter time series (P1,P2,P3) have lower discharges in Q_o^h and Q_m^h than the full period. The slight deterioration in in FHV for PFull is hence related to the very highest peak flows, which are more numerous in longer time series.

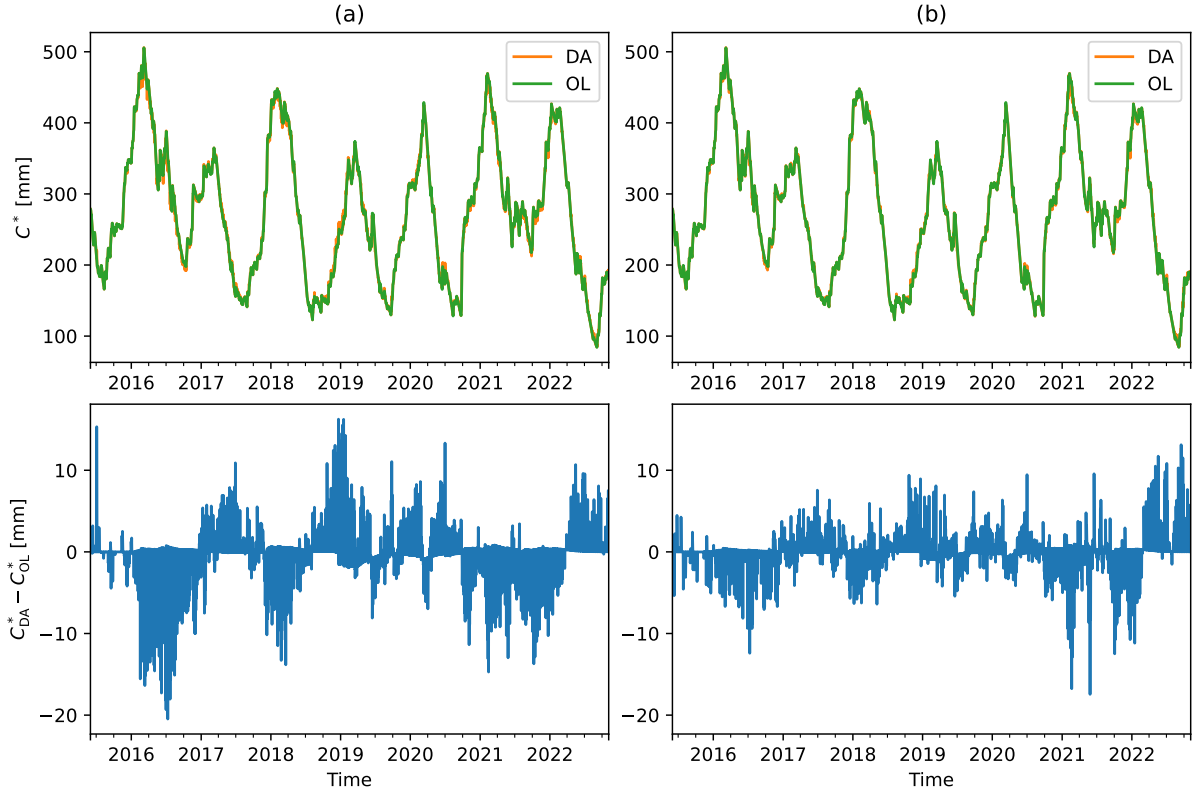

Figure 6.1: Top row: PDM C^* time series for DA (C^*_{DA}) and OL (C^*_{OL}). Bottom row: Difference between C^*_{DA} and C^*_{OL}. Column (a): LR full no forest as h^{-1}. Column (b): GPR full as h^{-1}.

To further investigate how the PDM is updated with DA, C^* and Q outputs of the PDM for both the best and the worst performing h^{-1} on PFull, LR full no forest and GPR full respectively, are given in Figure 6.1 (C^*) and 6.2 (Q). While hard to discern between C^* for DA (C^*_{DA}) and OL (C^*_{OL}) on the time series themselves, the difference between the two reveals that updates occur in the range of -20 to 15 mm for LR. For GPR, the changes in C^* after DA are smaller as C^*_{obs} is a closer fit to C^* from the PDM. This less pronounced update in C^* for GPR also translates to Q, with differences between DA (Q_{DA}) and OL (Q_{OL}) being both smaller and occurring less

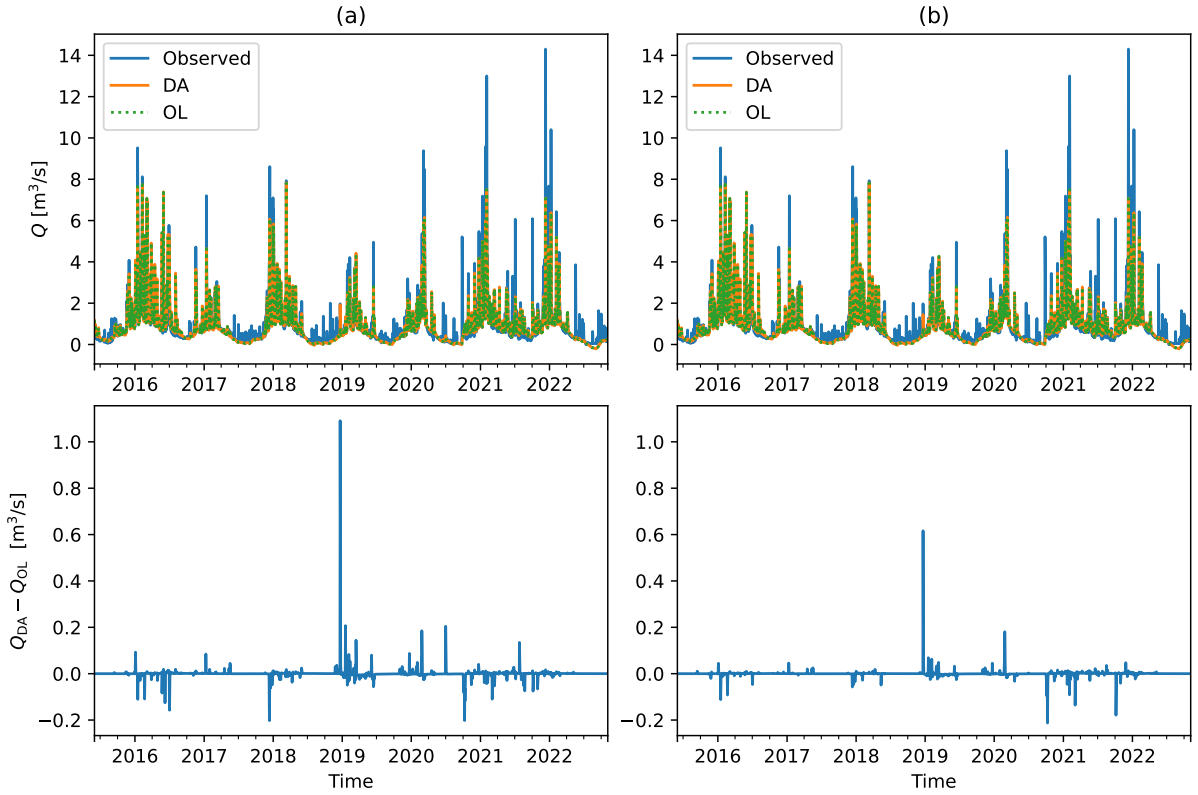

Figure 6.2: Top row: PDM Q time series for DA (Q_{DA}) and OL (Q_{OL}). Bottom row: Difference between Q_{DA} and Q_{OL}. Column (a): LR full no forest as h^{-1}. Column (b): GPR full as h^{-1}.

frequently than for LR. Both for LR and GPR however, the differences between Q_{DA} and Q_{OL} are very minor. When comparing Figure 6.2 with 6.1, it stands out that C^* being altered by DA, often does not lead to a change in Q. This seems to indicate a quite limited sensitivity of Q to C^* for updates of the magnitude as displayed here. For the other inverse observation operators, very similar patterns for C^* and Q are observed and are hence not further discussed.

Based on DA with the current parameters, it can be concluded that none of the h^{-1} methods yield significant improvements in flow prediction skill for the studied catchment. Furthermore, the use of more advanced nonlinear ML inverse observation operators does not seem of added value for the assimilation compared to the simpler linear methods.

6.4.2 Comparison of data assimilation parameters

Due to the only minor changes in predictive skill with DA regardless of the metric used for assessment, the parameter comparison will be based only on the ΔNSE. The results for the different γ and τ_a values for the five h^{-1} methods are displayed on Figure 6.3. As for the initial DA parameters, none of the parameter combinations

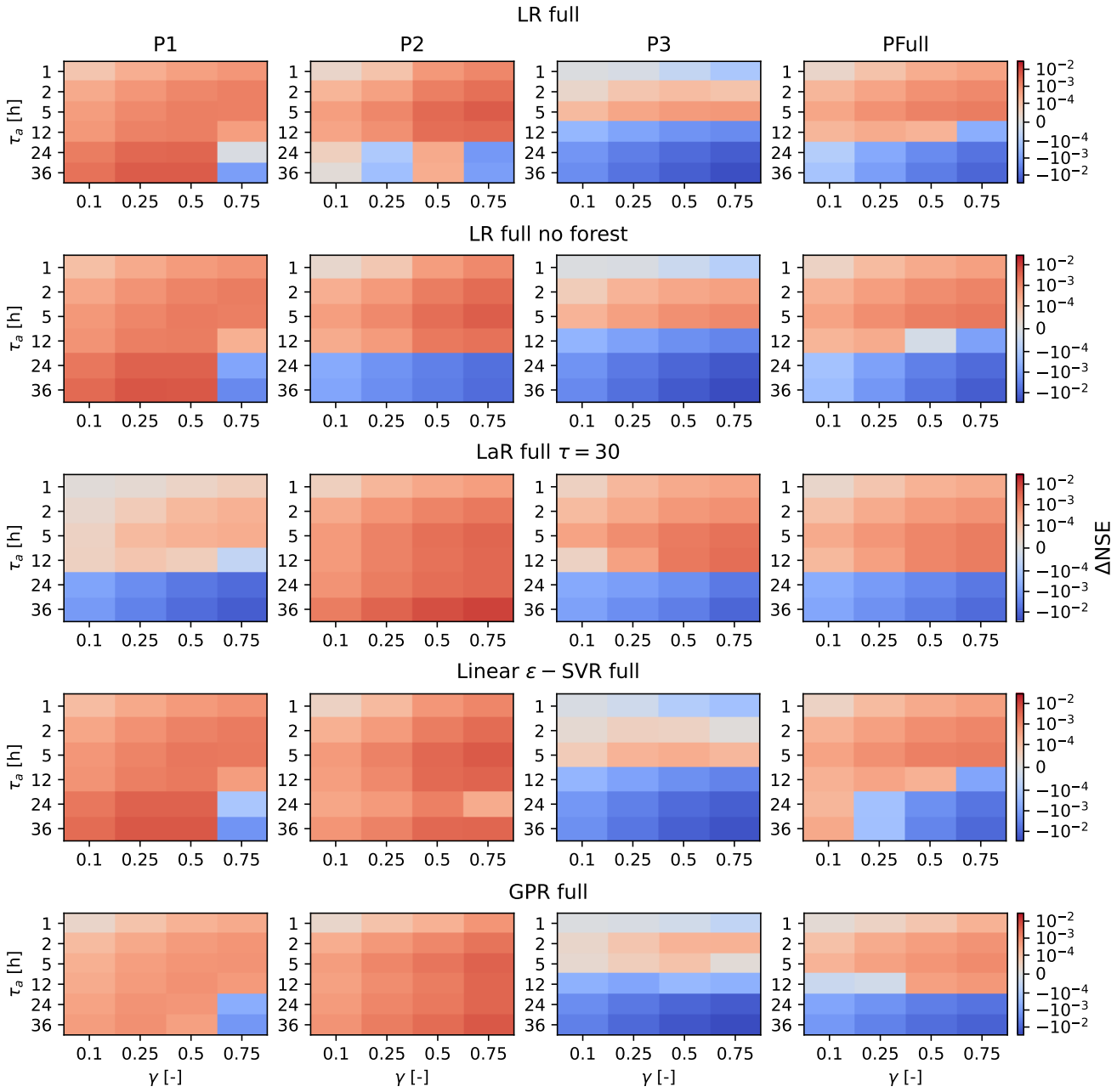

Figure 6.3: Color indicating ΔNSE in function of γ and τ_a parameters for DA. Columns indicate the four periods for evaluation, rows the different h^{-1} options.

yield large improvements in NSE, as the maximum ΔNSE over any of the periods is still < 0.01. Per period, following observations are made:

1. P1: For all h^{-1} methods except LaR, large values of τ_a (e.g. 24 or 36) and $\gamma \in \{0.25, 0.5\}$ give the largest ΔNSE.
2. P2: $\gamma = 0.75$ performs best combined with high τ_a (expect for LR).
3. P3: Lower τ_a, generally around 5 hours, with higher $\gamma \in \{0.5, 0.75\}$ yield the best results.
4. PFull: Similar trends as for P3.

The above indicates that when changing from periods for which the PDM and/or h^{-1} are calibrated/trained to periods of unseen data, a shorter time window for assimilation (and hence less influence of the observation) is preferred. A value of 0.5 seems to be a good choice for γ, but also 0.75 is suitable if a lower τ_a ($\leq$ 5 h) is chosen.

6.4.3 Discussion

As mentioned in Section 5.1, earlier work on using DA for the PDM with SSM products, retrieved from passive microwave radiometer satellites, has been performed by Alvarez-Garreton et al. (2014). Also Brocca et al. (2010) and Brocca et al. (2012) used similar SSM products for DA with other conceptual rainfall-runoff models. Contrary to the results obtained here, all of these works do obtain significant improvement in discharge prediction skill. For example, Alvarez-Garreton et al. (2014) obtain a 20 - 30 % improvement. One of the reasons why these earlier works might obtain better results, is that they transform the retrieved SSM with an exponential filter to the soil wetness index, which better represents the deeper SM layers. This is an important step, as the soil moisture reservoir of the PDM (but also of other conceptual models) represents SM layers significantly deeper than the SSM of the top few centimetres retrieved at C-band (cf. Section 3.1.2.3). In the current work, no transformation is applied on the γ^0 data to account for this.

Furthermore, the works of Brocca et al. (2012) and Alvarez-Garreton et al. (2014) use the more advanced ensemble Kalman filter. This has to advantage of providing a dynamic estimate of error on the modelled states. The resulting time-varying gain (or equivalently γ for Newtonian nudging), allows for better assessment of when to adhere more or less trust to modelled sates. Lastly, the methods relying on SSM assimilation have the advantage of using data coming from a retrieval model calibrated on fields measurements of SSM (see for example Owe et al. (2008)). Therefore the assimilation of SSM-derived observations (with a simple inverse observation operator), might be more reflective of actual observed SM conditions than when direct mapping from observation (i.e. backscatter) to model state is performed.

7. CONCLUSIONS AND FUTURE PERSPECTIVES

Volledig nieuw stuk. Indien mogelijk opnieuw, graag feedback van jullie beiden hier over. Ook een vraag hier: momenteel heb ik hier veel termen weer languit geschreven, is dit een goed idee of beter gewoon afkortingen blijven gebruiken?

The goal of this dissertation was to investigate the assimilation of SAR backscatter combined with an auxiliary vegetation index into a conceptual, lumped rainfall-runoff model for improved discharge predictions. For this purpose, several ML methods were assessed as potential inverse observation operator to map the remote sensing observations to a modelled state of the probability distributed model: the critical storage capacity C^*.

Before considering the outcomes of the data assimilation, the most prominent result from the calibration of the PDM is that for both optimisation techniques used (particle swarm optimisation and Nelder-Mead), underestimation of peak flows and bias on low baseflows occur. To reduce the presence of these unwanted features in the hydrograph, a possibility is the use of multi-objective calibration instead of the simpler single-objective calibration on the NSE used here. For example, Madsen (2000) proposes an objective function evaluating the water balance, shape of the hydrograph, low flows and peak flows.

Different ML methods, ranging from simple linear regression to the complex LSTM neural network, were used for the retrieval of C^*_{obs} from the backscatter and LAI data. Nearly all methods obtained retrievals very close to the C^* trainig targets with $R^2 > 0.8$ in the trainig period if the most extensive dataset was used. For the test data however, the linear methods showed significantly better generalisation than the non-linear ones. While the former had R^2_{test} values only slightly lower than R^2_{train} in most cases, this difference could go up to 0.3 for the latter. This overfitting is problematic, as retrievals too close to the modelled states will not lead to meaningful updates in assimilation.

The above was confirmed when performing data assimilation with the Newtonian nudging scheme. Although none of the inverse observation operators lead to significant improvement in NSE (ΔNSE < 0.01 regardless of the DA parameters used), non-linear methods yielded smaller updates than linear ones. This indicates that for the preliminary assessment on the Zwalm catchment, the hypothesis that using

more complex non-linear observation operators would improve predictive skill, is not confirmed.

Further research is needed however to more extensively asses the merit of using the ML observation operators. First, it should be investigated if updating of S_1 (as done in Alvarez-Garreton et al. (2014)) instead of C^* results in more significant changes in discharge prediction. Also a joint updating of C^*, S_1 and S_2 is of interest. Additionally, instead of using ML as retrieval algorithm, it could also be used as forward observation operator. Analogous to the work of Rains et al. (2022), C^* (and/or S_1) combined with LAI could be mapped to γ^0, which is then compared/combined with observed backscatter in sequential DA.

With regards to the DA method, a more correct assessment of the model and observational uncertainty in time should be investigated, as the current method uses a fixed factor to determine how much the modelled state is updated by the observation. This better error characterisation could be combined with the use of a more advanced DA technique such as the ensemble Kalman filter, as already used for SSM assimilation in the PDM by Alvarez-Garreton et al. (2014).

The most pressing question raised by this dissertation is not related to the DA methodology, but to why the more advanced ML techniques do not deliver superior performance. This can most likely be explained by a lack of data. Despite starting from a large RS dataset, fitting the big data definition of Reichstein et al. (2019), the spatial averaging per land use category significantly reduces the data volume to a dataset of around 600 trainig points with a dozen features only, which can be considered small in the context of ML.

A first possibility to increase the data volume, is the use of distributed instead instead of lumped rainfall-runoff models. If one (inverse) observation operator is trained for all grid cells, this would effectively increase the number of training points per time step from one to the number of grid cells. This is analogous to the approach proposed by Corchia et al. (2023), who combine modelled SSM (and ohter states) from a land surface model with LAI observations as inputs for an NN-based observation operator simulating σ^0 backscatter for use in DA.

It could also be attempted to train the inverse observation operator on several catchments at once, following the rational of Kratzert et al. (2019b) for LSTM-based rainfall-runoff modelling. The risk here is that due to different internal representation of states related to SM (e.g. related to the different parameters from calibration), learning a universal mapping from observation to states might be hindered. The limited sensitivity of the C^* Z-score with regards to different parameter sets shown in this dissertation, is a promising sign however for the potential of this concept.

Placing the research in a broader perspective, the above illustrates that the most important future research endeavour is the creation of large data sets over several catchments to fully asses the potential of not only the method proposed in this dissertation, but of all data-driven and hybrid approaches in rainfall-runoff modelling. This is confirmed by Nearing et al. (2021), who deem creating large standardised data repositories as the best investment in the field of machine learning for hydrology. Several countries such as the United States and Great Brittain already have these so-called Catchment Attributes and Meteorology for Large-sample Studies (CAMELS) datasets of standardised meteorological forcings, streamflow and static catchment attributes (Kratzert et al., 2023). The creation of such dataset for Flanders/Belgium is therefore of great interest. Combined with the spatial extension of the Sentinel-1 preprocessing in the OpenEO platform as proposed in this dissertation, one could truly test at large scale. The latter is paramount, as according to Nearing et al. (2021), deep learning methods do not scale like traditional methods. Consequently, small data sets (e.g. one catchment) do not tell much about the potential on large datasets. So although the hypothesis of this dissertation was not confirmed by the current approach, a clear vision for future research is obtained.

BIBLIOGRAPHY

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: large-scale machine learning on heterogeneous systems. Technical report, Google Brain. https://doi.org/10.48550/arXiv.1603.04467.

Alaska Satellite Facility (n.d.a). Introduction to SAR. https://hyp3-docs.asf.alaska.edu/guides/introduction_to_sar/. Accessed: 04/12/2022.

Alaska Satellite Facility (n.d.b). Sentinel-1 - data and imagery. https://asf.alaska.edu/data-sets/sar-data-sets/sentinel-1/sentinel-1-data-and-imagery/. Accessed: 02/12/2022.

Alvarez-Garreton, C., Ryu, D., Western, A., Crow, W., and Robertson, D. (2014). The impacts of assimilating satellite soil moisture into a rainfall–runoff model in a semi-arid catchment. *Journal of Hydrology*, 519:2763–2774. https://doi.org/10.1016/j.jhydrol.2014.07.041.

Aubert, D., Loumagne, C., and Oudin, L. (2003). Sequential assimilation of soil moisture and streamflow data in a conceptual rainfall–runoff model. *Journal of Hydrology*, 280(1):145–161. https://doi.org/10.1016/S0022-1694(03)00229-4.

Bergmeir, C. and Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213. https://doi.org/10.1016/j.ins.2011.12.028.

Beven, K. (2012a). *Rainfall-runoff modelling*, chapter 4: Predicting hydrographs using models based on data, pages 83–117. John Wiley & Sons, Ltd, Hoboken, NJ.

Beven, K. (2012b). *Rainfall-runoff modelling*, chapter 1: Down to basics: runoff processes and the modelling process, pages 1–23. John Wiley & Sons, Ltd, Hoboken, NJ.

Beven, K. (2012c). *Rainfall-runoff modelling*, chapter 7: Parameter estimation and predictive uncertainty, pages 231–287. John Wiley & Sons, Ltd, Hoboken, NJ.

Bishop, C. M. (2006a). *Pattern recognition and machine learning*, chapter 3: Linear models for regression, pages 137–177. Information science and statistics. Springer, New York, NY.

Bishop, C. M. (2006b). *Pattern recognition and machine learning*, chapter 7: Sparse kernel machines, pages 325–358. Information science and statistics. Springer, New York, NY.

Bishop, C. M. (2006c). *Pattern recognition and machine learning*, chapter 6: Kernel methods, pages 291–323. Information science and statistics. Springer, New York, NY.

Bishop, C. M. (2006d). *Pattern recognition and machine learning*, chapter 5: Neural networks, pages 225–290. Information science and statistics. Springer, New York, NY.

Bontje, D., Chapman, B., Dadamia, D., Kellndorfer, J., Killough, B., Labahn, S., Lavalle, M., Lewis, A., Metzger, M., Meyer, F., Miranda, N., Rosenqvist, A., Siqueira, A., Small, D., Tadono, T., Thankappan, M., Yuan, F., and Zhou, Z.-S. (2022). Normalised radar backscatter (CARD4L-NRB). Technical report, Committee on Earth Observation Satellites (CEOS).

Bouttier, F. and Courtier, P. (2002). Data assimilation concepts and methods. Meteorological training course lecture series, ECMWF.

Brocca, L., Melone, F., Moramarco, T., Wagner, W., Naeimi, V., Bartalis, Z., and Hasenauer, S. (2010). Improving runoff prediction through the assimilation of the ASCAT soil moisture product. *Hydrology and Earth System Sciences*, 14(10):1881–1893. https://doi.org/10.5194/hess-14-1881-2010.

Brocca, L., Moramarco, T., Melone, F., Wagner, W., Hasenauer, S., and Hahn, S. (2012). Assimilation of surface- and root-zone ASCAT soil moisture products into rainfall–runoff modeling. *IEEE Transactions on Geoscience and Remote Sensing*, 50(7):2542–2555. https://doi.org/10.1109/TGRS.2011.2177468.

Cabus, P. (2008). River flow prediction through rainfall–runoff modelling with a probability-distributed model (PDM) in Flanders, Belgium. *Agricultural Water Management*, 95(7):859–868. https://doi.org/10.1016/j.agwat.2008.02.013.

Cabus, P. (2021). IJkingskromess. Personal communication.

Canada Centre for Remote Sensing (2016). *Fundamentals of remote sensing*, chapter 2: Satellites and sensors, pages 34–91. Canada Centre for Mapping and Earth Observation, Ottawa, Canada.

CEOS (n.d.). CEOS analysis-ready datasets. https://ceos.org/ard/index.html#datasets. Accessed: 18/04/2023.

Chen, J. M. and Black, T. A. (1992). Defining leaf area index for non-flat leaves. *Plant, Cell & Environment*, 15(4):421–429. https://doi.org/10.1111/j.1365-3040.1992.tb00992.x.

Cheng, J. Y. and Mailund, T. (2015). Ancestral population genomics using coalescence hidden Markov models and heuristic optimisation algorithms. *Computational Biology and Chemistry*, 57:80–92. https://doi.org/10.1016/j.compbiolchem.2015.02.001.

Chollet, F. et al. (2015). Keras. https://keras.io. Accessed: 23/05/2023.

Copernicus (2022). Copernicus global 30 meter digital elevation model. https://doi.org/10.5270/ESA-c5d3d65. Accessed: 15/02/2023.

Copernicus (2023). Leaf area index. https://land.copernicus.eu/global/products/lai. Accessed: 21/02/2023.

Copernicus (n.d.). Infrastructure Overview. https://www.copernicus.eu/en/about-copernicus/infrastructure-overview. Accessed: 01/12/2022.

Corchia, T., Bonan, B., Rodriguez-Fernandez, N., Colas, G., and Calvet, J.-C. (2023). Added value of machine learning in the assimilation of ASCAT observations into the ISBA land surface model. Vienna, Austria. EGU General Assembly 2023. https://doi.org/10.5194/egusphere-egu23-6976.

De Lannoy, G. J. M., Bechtold, M., Albergel, C., Brocca, L., Calvet, J.-C., Carrassi, A., Crow, W. T., de Rosnay, P., Durand, M., Forman, B., Geppert, G., Girotto, M., Hendricks Franssen, H.-J., Jonas, T., Kumar, S., Lievens, H., Lu, Y., Massari, C., Pauwels, V. R. N., Reichle, R. H., and Steele-Dunne, S. (2022). Perspective on satellite-based land data assimilation to estimate water cycle components in an era of advanced data availability and model sophistication. *Frontiers in Water*, 4. https://doi.org/10.3389/frwa.2022.981745.

Departement Omgeving (2023). Landgebruik - Vlaanderen - toestand 2019. https://www.vlaanderen.be/datavindplaats/catalogus/landgebruik-vlaanderen-toestand-2019. Accessed: 16/02/2023.

Dewelde, J., Verbeke, S., Quintelier, E., Cabus, P., Vermeulen, A., Vansteenkiste, T., de Jongh, I., and Cauwenberghs, K. (2014). Real-time flood forecasting systems in Flanders. New York, NY. 11th International Conference on Hydroinformatics. https://academicworks.cuny.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1218&context=cc_conf_hic.

Dormann, C. F., Elith, J., Bacher, S., Buchmann, C., Carl, G., Carré, G., Marquéz, J. R. G., Gruber, B., Lafourcade, B., Leitão, P. J., Münkemüller, T., McClean, C., Osborne, P. E., Reineking, B., Schröder, B., Skidmore, A. K., Zurell, D., and Lautenbach, S. (2013). Collinearity: a review of methods to deal with it

and a simulation study evaluating their performance. *Ecography*, 36(1):27–46. https://doi.org/10.1111/j.1600-0587.2012.07348.x.

Dostálová, A., Wagner, W., Milenković, M., and Hollaus, M. (2018). Annual seasonality in Sentinel-1 signal for forest mapping and forest type classification. *International Journal of Remote Sensing*, 39(21):7738–7760. https://doi.org/10.1080/01431161.2018.1479788.

Dries, J. and Van Tricht, K. (2019). Terrascope Sentinel-1 algorithm theoretical base document S1 – Sigma0 GRD V110. Technical report, VITO.

Duan, Q., Gupta, V. K., and Sorooshian, S. (1993). Shuffled complex evolution approach for effective and efficient global minimization. *Journal of Optimization Theory and Applications*, 76:501–521. https://doi.org/10.1007/BF00939380.

El Khaled, D., Castellano, N. N., Gázquez, J. A., Perea-Moreno, A.-J., and Manzano-Agugliaro, F. (2016). Dielectric spectroscopy in biomaterials: Agrophysics. *Materials*, 9(5):310. https://doi.org/10.3390/ma9050310.

Engelbrecht, A. (2007). *Computational intelligence: an introduction*, chapter 16: Particle swarm optimization, pages 289–358. John Wiley & Sons, Ltd, Hoboken, NJ.

ESA (n.d.a). Sentinel-1. https://sentinel.esa.int/web/sentinel/missions/sentinel-1. Accessed: 01/12/2022.

ESA (n.d.b). Sentinel-1 SAR user guide. https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel-1-sar. Accessed: 01/12/2022.

Evensen, G., Vossepoel, F. C., and van Leeuwen, P. J. (2022). *Data assimilation fundamentals: a unified formulation of the state and parameter estimation problem*, chapter 1: Introduction, pages 1–5. Springer textbooks in earth sciences, geography and environment. Springer Nature, Cham, Switzerland.

Feng, S., Zhou, H., and Dong, H. (2019). Using deep neural network with small dataset to predict material defects. *Materials & Design*, 162:300–310. https://doi.org/10.1016/j.matdes.2018.11.060.

Gao, F. and Han, L. (2012). Implementing the Nelder-Mead simplex algorithm with adaptive parameters. *Computational Optimization and Applications*, 51(1):259–277. https://doi.org/10.1007/s10589-010-9329-3.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, M., editors, *Proceedings of the thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of machine learning research*, pages 249–256, Sardinia, Italy. PMLR. https://proceedings.mlr.press/v9/glorot10a.html.

Goodfellow, I. J., Bengio, Y., and Courville, A. (2016a). *Deep learning*, chapter 6: Deep feedforward networks, pages 168–227. MIT Press, Cambridge, MA. http://www.deeplearningbook.org.

Goodfellow, I. J., Bengio, Y., and Courville, A. (2016b). *Deep learning*, chapter 8: Optimization for training deep models, pages 274–329. MIT Press, Cambridge, MA. http://www.deeplearningbook.org.

Goodfellow, I. J., Bengio, Y., and Courville, A. (2016c). *Deep learning*, chapter 7: Regularization for deep learning, pages 228–272. MIT Press, Cambridge, MA. http://www.deeplearningbook.org.

Han, S., Qubo, C., and Meng, H. (2012). Parameter selection in SVM with RBF kernel function. In *World Automation Congress 2012*, Proceedings of the Biannual World Automation Congress, pages 1–4, Puerto Vallarta, Mexico. IEEE. https://ieeexplore.ieee.org/abstract/document/6321759.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362. https://doi.org/10.1038/s41586-020-2649-2.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference and prediction*, chapter 3: Linear methods for regression, pages 43–99. Springer, New York, NY, Second edition. https://hastie.su.domains/ElemStatLearn/.

Hoedt, P.-J., Kratzert, F., Klotz, D., Halmich, C., Holzleitner, M., Nearing, G. S., Hochreiter, S., and Klambauer, G. (2021). MC-LSTM: Mass-conserving LSTM. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of machine learning research*, pages 4275–4286. PMLR. https://proceedings.mlr.press/v139/hoedt21a.html.

Höge, M., Scheidegger, A., Baity-Jesi, M., Albert, C., and Fenicia, F. (2022). Improving hydrologic models for predictions and process understanding using neural ODEs. *Hydrology and Earth System Sciences*, 26(19):5085–5102. https://doi.org/10.5194/hess-26-5085-2022.

Hostache, R., Rains, D., Mallick, K., Chini, M., Pelich, R., Lievens, H., Fenicia, F., Corato, G., Verhoest, N. E. C., and Matgen, P. (2020). Assimilation of Soil Moisture and Ocean Salinity (SMOS) brightness temperature into a large-scale distributed conceptual hydrological model to improve soil moisture predictions: the Murray–Darling basin in Australia as a test case. *Hydrology and Earth System Sciences*, 24(10):4793–4812. https://doi.org/10.5194/hess-24-4793-2020.

Houser, P. R., De Lannoy, G. J., and Walker, J. P. (2010). Land surface data assimilation. In Lahoz, W., Khattatov, B., and Menard, R., editors, *Data assimilation: making sense of observations*, pages 549–597. Springer Berlin Heidelberg, Heidelberg, Germany.

Houser, P. R., Shuttleworth, W. J., Famiglietti, J. S., Gupta, H. V., Syed, K. H., and Goodrich, D. C. (1998). Integration of soil moisture remote sensing and hydrologic modeling using data assimilation. *Water Resources Research*, 34(12):3405–3420. https://doi.org/10.1029/1998WR900001.

Hoyer, S. and Hamman, J. (2017). xarray: N-D labeled arrays and datasets in Python. *Journal of Open Research Software*, 5(1). https://doi.org/10.5334/jors.148.

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95. https://doi.org/10.1109/MCSE.2007.55.

Hyndman, R. J. and Fan, Y. (1996). Sample quantiles in statistical packages. *The American Statistician*, 50(4):361–365. https://doi.org/10.2307/2684934.

Ide, K., Courtier, P., Ghil, M., and Lorenc, A. C. (1997). Unified notation for data assimilation: Operational, sequential and variational. *Journal of the Meteorological Society of Japan. Ser. II*, 75(1B):181–189. https://doi.org/10.2151/jmsj1965.75.1B_181.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2021). *An introduction to statistical learning: with applications in R*, chapter 5: Resampling methods, pages 197–223. Springer, New York, NY, Second edition.

Jiang, S., Zheng, Y., and Solomatine, D. (2020). Improving AI system awareness of geoscience knowledge: symbiotic integration of physical approaches and deep learning. *Geophysical Research Letters*, 47(13):e2020GL088229. https://doi.org/10.1029/2020GL088229.

Jordahl, K., Van den Bossche, J., Fleischmann, M., Wasserman, J., McBride, J., Gerard, J., Tratner, J., Perry, M., Badaracco, A. G., Farmer, C., Hjelle, G. A., Snow, A. D., Cochran, M., Gillies, S., Culbertson, L., Bartos, M., Eubank, N., maxalbert, Bilogur, A., Rey, S., Ren, C., Arribas-Bel, D., Wasser, L., Wolf, L. J., Journois, M., Wilson, J., Greenhall, A., Holdgraf, C., Filipe, and Leblanc, F. (2022). geopandas/geopandas: v0.11.1. https://doi.org/10.5281/zenodo.6894736.

Kalnay, E. (2002). *Atmospheric modeling, data assimilation and predictability*, chapter 5: Data assimilation, pages 136–204. Cambridge University Press, Cambridge, United Kingdom.

Kamali, B., Mousavi, S. J., and Abbaspour, K. (2013). Automatic calibration of HEC-HMS using single-objective and multi-objective PSO algorithms. *Hydrological Processes*, 27(26):4028–4042. https://doi.org/10.1002/hyp.9510.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4 of *International Conference on Neural Networks*, pages 1942–1948, Perth, Australia. IEEE. https://doi.org/10.1109/ICNN.1995.488968.

Kingma, D. P. and Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. https://doi.org/10.48550/arXiv.1412.6980.

Kratzert, F., Herrnegger, M., Klotz, D., Hochreiter, S., and Klambauer, G. (2019a). NeuralHydrology – interpreting LSTMs in hydrology. In *Explainable AI: interpreting, explaining and visualizing deep learning*, Lecture notes in artifical intelligence, pages 347–362. Springer Nature, Charm, Switzerland.

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., and Herrnegger, M. (2018). Rainfall–runoff modelling using Long Short-Term Memory (LSTM) networks. *Hydrology and Earth System Sciences*, 22(11):6005–6022. https://doi.org/10.5194/hess-22-6005-2018.

Kratzert, F., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., and Nearing, G. (2019b). Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *Hydrology and Earth System Sciences*, 23(12):5089–5110. https://doi.org/10.5194/hess-23-5089-2019.

Kratzert, F., Nearing, G., Addor, N., Erickson, T., Gauch, M., Gilon, O., Gudmundsson, L., Hassidim, A., Klotz, D., Nevo, S., et al. (2023). Caravan - a global community dataset for large-sample hydrology. *Scientific Data*, 10(1):61. https://doi.org/10.1038/s41597-023-01975-w.

Landuyt, L., Verhoest, N., and Vancoillie, F. (2021). *Flood mapping from radar remote sensing using automated image classification techniques*. PhD thesis, Ghent University, Ghent, Belgium. ISBN: 9789463574150.

Legates, D. R. and McCabe Jr., G. J. (1999). Evaluating the use of "goodness-of-fit" measures in hydrologic and hydroclimatic model validation. *Water Resources Research*, 35(1):233–241. https://doi.org/10.1029/1998WR900018.

Liang, J., Terasaki, K., and Miyoshi, T. (2023). A machine learning approach to the observation operator for satellite radiance data assimilation. *Journal of the Meteorological Society of Japan*, 101(1):79–95. https://doi.org/10.2151/jmsj.2023-005.

Lievens, H. (2023). MLP structuur. Personal communication.

Lillesand, T., Kiefer, R., and Chipman, J. (2015). *Remote sensing and image interpretation*, chapter 6: Microwave and lidar sensing, pages 385–484. John Wiley & Sons, Ltd, Hoboken, NJ, Seventh edition.

Lindström, G., Johansson, B., Persson, M., Gardelin, M., and Bergström, S. (1997). Development and test of the distributed HBV-96 hydrological model. *Journal of Hydrology*, 201(1):272–288. https://doi.org/10.1016/S0022-1694(97)00041-3.

Liu, Y., Weerts, A. H., Clark, M., Hendricks Franssen, H.-J., Kumar, S., Moradkhani, H., Seo, D.-J., Schwanenberg, D., Smith, P., van Dijk, A. I. J. M., van Velzen, N., He, M., Lee, H., Noh, S. J., Rakovec, O., and Restrepo, P. (2012). Advancing data assimilation in operational hydrologic forecasting: progresses, challenges, and emerging opportunities. *Hydrology and Earth System Sciences*, 16(10):3863–3887. https://doi.org/10.5194/hess-16-3863-2012.

Madsen, H. (2000). Automatic calibration of a conceptual rainfall–runoff model using multiple objectives. *Journal of Hydrology*, 235(3):276–288. https://doi.org/10.1016/S0022-1694(00)00279-1.

Maroy, E., Velez, C., Pereira, F., Nossent, J., and Mostaert, F. (2021). Modelling water availability and water allocation strategies in the Scheldt basin: Sub report 4-3 – Analyses of hydrological models for climate change modelling – PDM modelling. Technical Report 2.0, Flanders Hydraulics Research.

Martens, B., Miralles, D., Lievens, H., Fernández-Prieto, D., and Verhoest, N. (2016). Improving terrestrial evaporation estimates over continental Australia through assimilation of SMOS soil moisture. *International Journal of Applied Earth Observation and Geoinformation*, 48:146–162. https://doi.org/10.1016/j.jag.2015.09.012.

Meyer, F. (2019). Spaceborne synthetic aperture radar: principles, data access, and basic processing techniques. In *The SAR handbook: comprehensive methodologies for forest monitoring and biomass estimation*, pages 21–62. SERVIR, Huntsville, AL.

Miralles, D. G., Brutsaert, W., Dolman, A. J., and Gash, J. H. (2020). On the use of the term “evapotranspiration”. *Water Resources Research*, 56(11):e2020WR028055. https://doi.org/10.1029/2020WR028055.

Miranda, L. J. V. (2018). PySwarms, a research-toolkit for particle swarm optimization in Python. *Journal of Open Source Software*, 3(21):433. https://doi.org/10.21105/joss.00433.

Moler, C. B. (2004). *Numerical computing with MATLAB*, chapter 3: Interpolation, pages 93–116. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Monteith, J. L. (1965). Evaporation and environment. *Symposia of the Society for Experimental Biology*, 19:205–234. https://repository.rothamsted.ac.uk/item/8v5v7/evaporation-and-environment.

Moore, R. J. (2007). The PDM rainfall-runoff model. *Hydrology and Earth System Sciences*, 11(1):483–499. https://doi.org/10.5194/hess-11-483-2007.

Moriasi, D. N., Gitau, M. W., Pai, N., and Daggupati, P. (2015). Hydrologic and water quality models: performance measures and evaluation criteria. *Transactions of the ASABE*, 58(6):1763–1785. https://doi.org/10.13031/trans.58.10715.

Nash, J. and Sutcliffe, J. (1970). River flow forecasting through conceptual models part I — A discussion of principles. *Journal of Hydrology*, 10(3):282–290. https://doi.org/10.1016/0022-1694(70)90255-6.

Nearing, G. S., Kratzert, F., Sampson, A. K., Pelissier, C. S., Klotz, D., Frame, J. M., Prieto, C., and Gupta, H. V. (2021). What role does hydrological science play in the age of machine learning? *Water Resources Research*, 57(3):e2020WR028091. https://doi.org/10.1029/2020WR028091.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, 7(4):308–313. https://doi.org/10.1093/comjnl/7.4.308.

Ng, A. Y. (1997). Preventing "overfitting" of cross-validation data. In *Proceedings of the fourteenth International Conference on Machine Learning*, volume 97, pages 245–253, Nashville, TN. ACM.

Olah, C. (2015). Understanding LSTM Networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs/. Accessed: 07/07/2023.

openEO Platform (2023). CARD4L NRB for SENTINEL1 GRD collection (provided by Sentinel Hub). https://docs.openeo.cloud/usecases/ard/sar/#card4l-nrb-for-sentinel1-grd-collection-provided-by-sentinel-hub. Accessed: 18/04/2023.

Owe, M., de Jeu, R., and Holmes, T. (2008). Multisensor historical climatology of satellite-derived global land surface moisture. *Journal of Geophysical Research: Earth Surface*, 113(F01002). https://doi.org/10.1029/2007JF000769.

Owe, M. and Van de Griend, A. A. (1998). Comparison of soil moisture penetration depths for several bare soils at two microwave frequencies and implications for remote sensing. *Water Resources Research*, 34(9):2319–2327. https://doi.org/10.1029/98WR01469.

Paniconi, C., Marrocu, M., Putti, M., and Verbunt, M. (2003). Newtonian nudging for a Richards equation-based distributed hydrological model. *Advances in Water Resources*, 26(2):161–178. https://doi.org/10.1016/S0309-1708(02)00099-4.

Pauwels, V. R. N., Verhoest, N. E. C., and De Troch, F. P. (2002). A metahillslope model based on an analytical solution to a linearized Boussinesq equation for temporally variable recharge rates. *Water Resources Research*, 38(12):33–1–33–14. https://doi.org/10.1029/2001WR000714.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830. https://doi.org/10.48550/arXiv.1201.0490.

Perrin, C., Michel, C., and Andréassian, V. (2003). Improvement of a parsimonious model for streamflow simulation. *Journal of Hydrology*, 279(1):275–289. https://doi.org/10.1016/S0022-1694(03)00225-7.

Piotrowski, A. P., Napiorkowski, J. J., and Piotrowska, A. E. (2020). Population size in particle swarm optimization. *Swarm and Evolutionary Computation*, 58:100718. https://doi.org/10.1016/j.swevo.2020.100718.

Plaza Guingla, D. A., De Keyser, R., De Lannoy, G. J. M., Giustarini, L., Matgen, P., and Pauwels, V. R. N. (2013). Improving particle filters in rainfall-runoff models: application of the resample-move step and the ensemble Gaussian particle filter. *Water Resources Research*, 49(7):4005–4021. https://doi.org/10.1002/wrcr.20291.

Project Jupyter, Matthias Bussonnier, Jessica Forde, Jeremy Freeman, Brian Granger, Tim Head, Chris Holdgraf, Kyle Kelley, Gladys Nalvarte, Andrew Osheroff, Pacer, M., Yuvi Panda, Fernando Perez, Benjamin Ragan Kelley, and Carol Willing (2018). Binder 2.0 - reproducible, interactive, sharable environments for science at scale. In Fatih Akici, David Lippa, Dillon Niederhut, and Pacer, M., editors, *Proceedings of the 17th Python in Science Conference*, pages 113 – 120, Austin, TX. SciPy. https://doi.org/10.25080/Majora-4af1f417-011.

Rains, D., Lievens, H., De Lannoy, G. J. M., Mccabe, M. F., de Jeu, R. A. M., and Miralles, D. G. (2022). Sentinel-1 backscatter assimilation using support vector regression or the water cloud model at European soil moisture sites. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5. https://doi.org/10.1109/LGRS.2021.3073484.

Ranjan, C. (2020a). *Understanding deep learning application in rare event prediction*, chapter 4: Multi-layer perceptrons, pages 39–106. USA.

Ranjan, C. (2020b). *Understanding deep learning application in rare event prediction*, chapter 5: LSTM layer and network structure, pages 107–167. USA.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for machine learning*, chapter 2: Regression, pages 7–31. Adaptive computation and machine learning. MIT Press, Cambrdige, MA.

Razavi, S. (2021). Deep learning, explained: fundamentals, explainability, and bridgeability to process-based modelling. *Environmental Modelling & Software*, 144:105159. https://doi.org/10.1016/j.envsoft.2021.105159.

Reichle, R. H. (2008). Data assimilation methods in the earth sciences. *Advances in Water Resources*, 31(11):1411–1418. https://doi.org/10.1016/j.advwatres.2008.01.001.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., and Prabhat (2019). Deep learning and process understanding for data-driven earth system science. *Nature*, 566:195–204. https://doi.org/10.1038/s41586-019-0912-1.

Rosenqvist, A. and Killough, B. (2018). A layman's interpretation guide to L-band and C-band synthetic aperture radar data, v2.0. Technical report, Committee on Earth Observation Satellites (CEOS).

Rubanova, Y., Chen, R. T. Q., and Duvenaud, D. K. (2019). Latent ordinary differential equations for irregularly-sampled time series. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in neural information processing systems 32 (NeurIPS 2019)*, volume 32. Neural Information Processing Systems Foundation. https://doi.org/10.48550/arXiv.1907.03907.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215. https://doi.org/10.1038/s42256-019-0048-x.

Schramm, M., Pebesma, E., Milenković, M., Foresta, L., Dries, J., Jacob, A., Wagner, W., Mohr, M., Neteler, M., Kadunc, M., Miksa, T., Kempeneers, P., Verbesselt, J., Gößwein, B., Navacchi, C., Lippens, S., and Reiche, J. (2021). The openEO API–harmonising the use of earth observation cloud services using virtual data cube functionalities. *Remote Sensing*, 13(6):1125. https://doi.org/10.3390/rs13061125.

Sentinel Hub (2023). Sentinel-1 GRD: processing chain. https://docs.sentinel-hub.com/api/latest/data/sentinel-1-grd/#processing-chain. Accessed: 18/04/2023.

Service public de Wallonie (2022). Occupation du sol en Wallonie - WALOUS 2018. https://geoportail.wallonie.be/catalogue/a0ad23a1-1845-4bd5-8c2f-0f62d3f1ec75.html. Accessed: 20/04/2023.

Shi, Y. (2004). Particle swarm optimization. *IEEE Connections*, 2(1):8–13. https://www.marksmannet.com/RobertMarks/Classes/ENGR5358/Papers/pso_bySHI.pdf.

Shi, Y. and Eberhart, R. (1999). Empirical study of particle swarm optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation- CEC99*, volume 3 of *Congress on Evolutionary Computation*, pages 1945–1950, Washington, DC. IEEE. https://doi.org/10.1109/CEC.1999.785511.

Siyang, J. (2019). Data assimilation with a machine learned observation operator and application to the assimilation of satellite data for sea ice models. Master's thesis, University of North Carolina at Chapel Hill, Chapel Hill, NC.

Small, D. (2011). Flattening gamma: radiometric terrain correction for SAR imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 49(8):3081–3093. https://doi.org/10.1109/TGRS.2011.2120616.

Smets, K., Verdonk, B., and Jordaan, E. M. (2007). Evaluation of performance measures for SVR hyperparameter selection. In *2007 International Joint Conference on Neural Networks*, International Joint Conference on Neural Networks (IJCNN), pages 637–642, Orlando, FL. IEEE. https://doi.org/10.1109/IJCNN.2007.4371031.

Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88.

Solomatine, D. and Wagener, T. (2011). 2.16 - Hydrological modeling. In Wilderer, P., editor, *Treatise on water science*, pages 435–457. Elsevier, Amsterdam, The Netherlands.

Stagge, J. H., Rosenberg, D. E., Abdallah, A. M., Akbar, H., Attallah, N. A., and James, R. (2019). Assessing data availability and research reproducibility in hydrology and water resources. *Scientific Data*, 6(1):190030. https://doi.org/10.1038/sdata.2019.30.

Staudemeyer, R. C. and Morris, E. R. (2019). Understanding LSTM – a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*. https://doi.org/10.48550/arXiv.1909.09586.

Stauffer, D. R. and Seaman, N. L. (1990). Use of four-dimensional data assimilation in a limited-area mesoscale model. Part I: experiments with synoptic-scale data. *Monthly Weather Review*, 118(6):1250 – 1277. https://doi.org/10.1175/1520-0493(1990)118<1250:UOFDDA>2.0.CO;2.

Tayfur, G. (2017). Modern optimization methods in water resources planning, engineering and management. *Water Resources Management*, 31:3205–3233. https://doi.org/10.1007/s11269-017-1694-6.

TensorFlow (2022). Time series forecasting. https://www.tensorflow.org/tutorials/structured_data/time_series. Accessed: 08/05/2023.

Thiessen, A. H. (1911). Precipitation averages for large areas. *Monthly Weather Review*, 39(7):1082 – 1089. https://doi.org/10.1175/1520-0493(1911)39<1082b:PAFLA>2.0.CO;2.

Van Hoey, S., Van De Wauw, J., Maiheau, B., and Buekenhout, D. (2021). pywaterinfo. https://fluves.github.io/pywaterinfo/index.html. Accessed: 23/05/2023.

Van Rossum, G. and Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.

Vernieuwe, H., Georgieva, O., De Baets, B., Pauwels, V. R. N., and Verhoest, N. E. C. (2003). Fuzzy models of rainfall-discharge dynamics. In Bilgiç, T., De Baets, B., and Kaynak, O., editors, *Fuzzy sets and systems — IFSA 2003*, Lecture notes in computer science, pages 303–310, Istanbul, Turkey. Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-44967-1_36.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272. https://doi.org/10.1038/s41592-019-0686-2.

VITO (n.d.). Sentinel-1 data products. https://docs.terrascope.be/#/DataProducts/Sentinel-1/ProductsOverview?id=sentinel-1-data-products. Accessed: 23/02/2023.

Vlaamse Milieumaatschappij (2022). Vlaamse Hydrografische Atlas. https://www.vlaanderen.be/datavindplaats/catalogus/vlaamse-hydrografische-atlas-waterlopen-3-maart-2022. Accessed: 23/05/2023.

Vogel, R. M. and Fennessey, N. M. (1994). Flow-duration curves. I: new interpretation and confidence intervals. *Journal of Water Resources Planning and Management*, 120(4):485–504. https://doi.org/10.1061/(ASCE)0733-9496(1994)120:4(485).

Wagener, T., Sivapalan, M., Troch, P. A., McGlynn, B. L., Harman, C. J., Gupta, H. V., Kumar, P., Rao, P. S. C., Basu, N. B., and Wilson, J. S. (2010). The future of hydrology: an evolving science for a changing world. *Water Resources Research*, 46(5):W05301. https://doi.org/10.1029/2009WR008906.

Weerakody, P. B., Wong, K. W., Wang, G., and Ela, W. (2021). A review of irregular time series data handling with gated recurrent neural networks. *Neurocomputing*, 441:161–178. https://doi.org/10.1016/j.neucom.2021.02.046.

Wolfs, D., Verger, A., Van der Goten, R., and Sanchez-Zapero, J. (2022). Product user manual: Leaf area index (LAI), fraction of absorbed photosynthetically active

radiation (FAPAR), fraction of green vegetation cover (FCover), collection 300m version 1.1. Technical report, Copernicus Global Land Operations.

Woodhouse, I. (2006a). *Introduction to microwave remote sensing*, chapter 1: Why microwaves, pages 1–5. CRC Press, Boca Raton, FL.

Woodhouse, I. (2006b). *Introduction to microwave remote sensing*, chapter 10: Imaging radar, pages 259–303. CRC Press, Boca Raton, FL.

Woodhouse, I. (2006c). *Introduction to microwave remote sensing*, chapter 5: Microwaves in the real word, pages 93–149. CRC Press, Boca Raton, FL.

Yilmaz, K. K., Gupta, H. V., and Wagener, T. (2008). A process-based diagnostic approach to model evaluation: application to the NWS distributed hydrologic model. *Water Resources Research*, 44(9):W09417. https://doi.org/10.1029/2007WR006716.

Yu, Y., Si, X., Hu, C., and Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation*, 31(7):1235–1270. https://doi.org/10.1162/neco_a_01199.

APPENDIX A

SUPPLEMENTARY MATERIAL

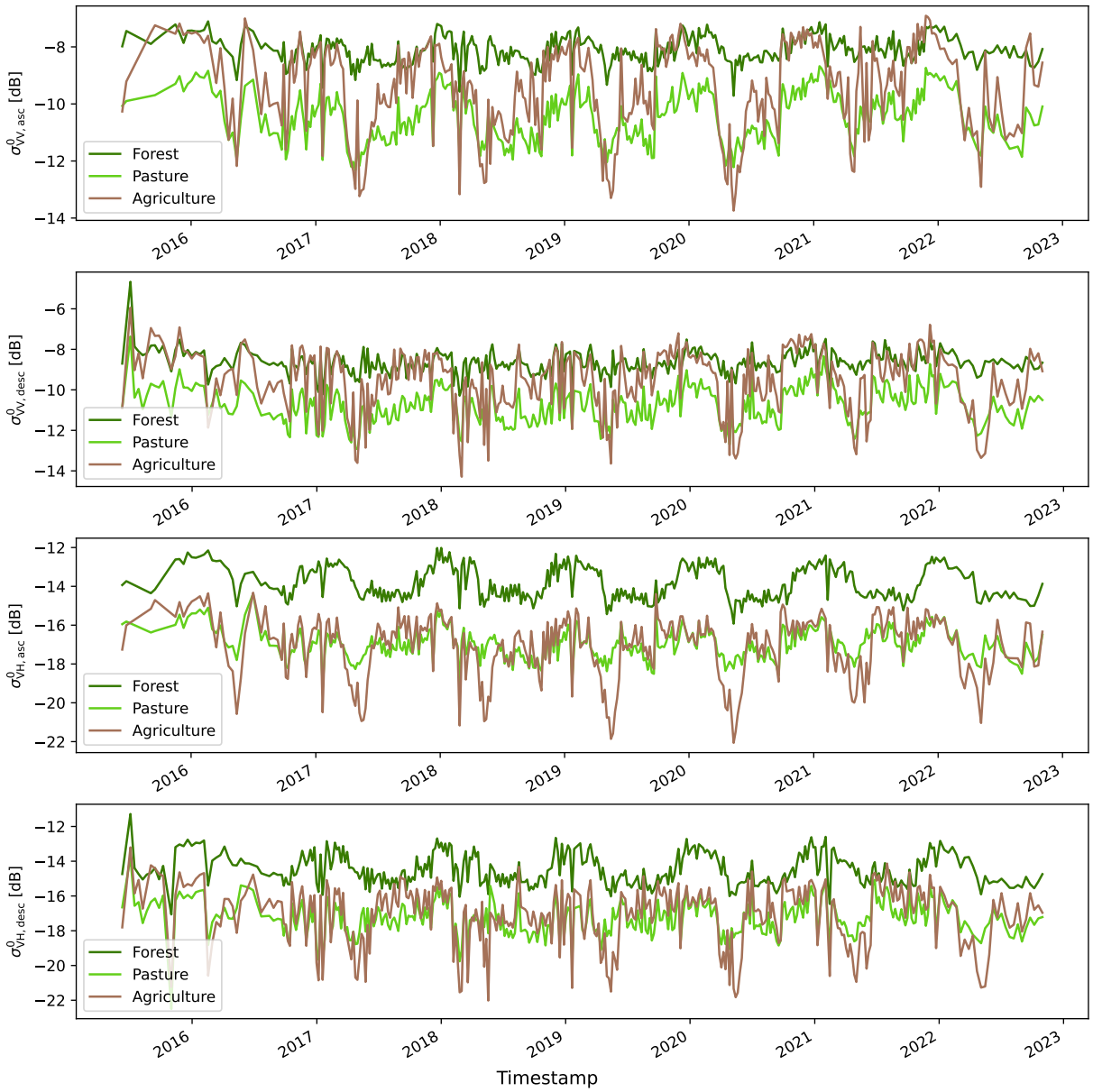

Figure A.1: Time series for σ^0 in (from top to bottom) VV with ascending orbit, VV with descending orbit, VH with ascending orbit and VH with descending orbit

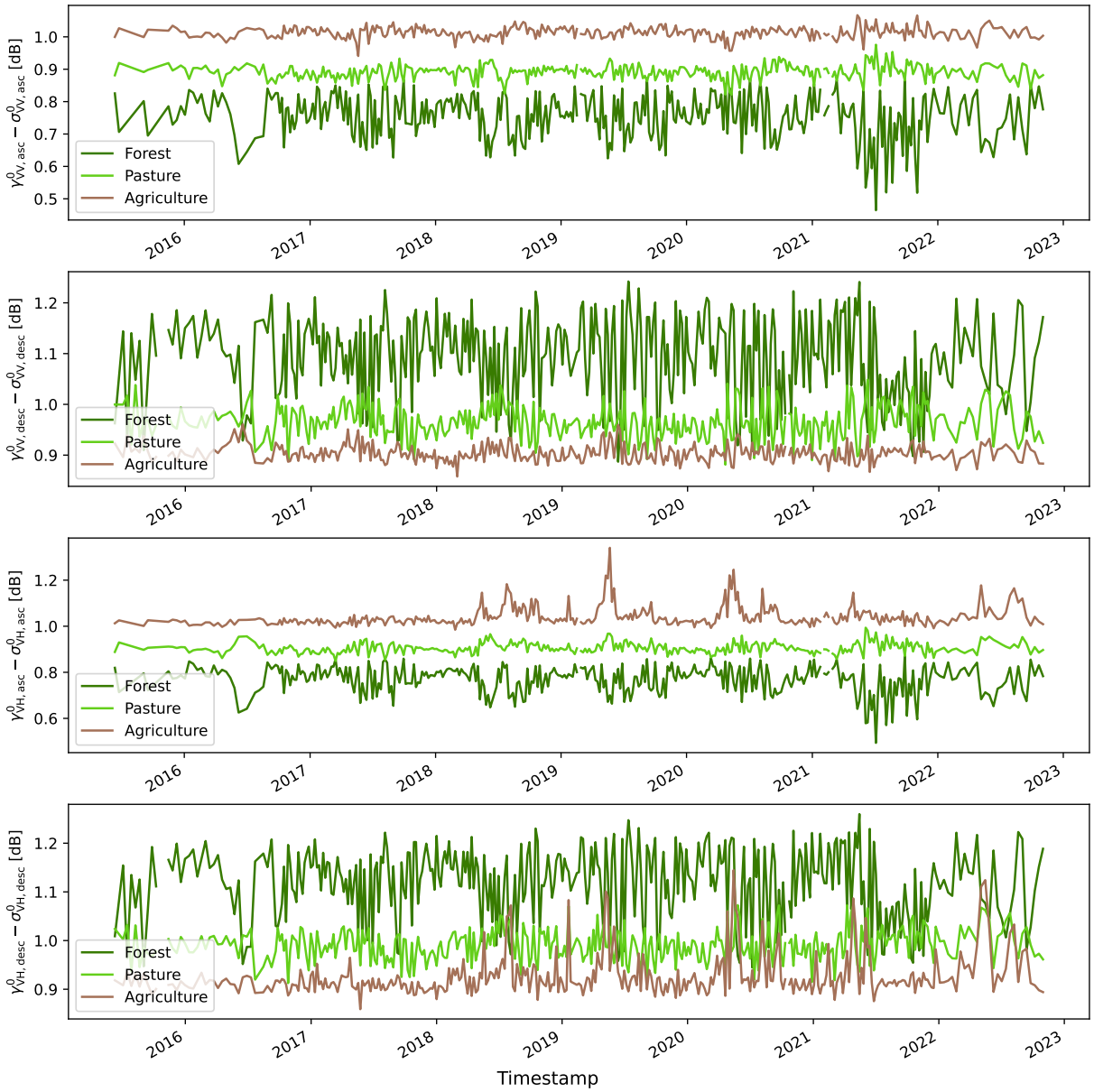

Figure A.2: Difference between γ^0_T and σ^0 times series for VV with ascending orbit, VV with descending orbit, VH with ascending orbit and VH with descending orbit.

Table A.1: Parameter values of the PDM for the initial, NM optimal and PSO optimal and parameter set. Minimum and maximum values are determined from Cabus (2008) and Cabus (2021).

Parameter	Min	Max	Initial	NM optimal	PSO optimal
Area [km^2]	-	-	109.23	115.21	115.21
Soil moisture storage					
c_{max} [mm]	160	5000	400.61	1244.01	698.13
c_{min}[mm]	0	300	86.68	220.03	187.01
b [-]	0.1	2	0.6	2	0.54
Evaporation					
b_e [-]	1	3	3	3	2.97
Groundwater recharge					
k_g [h mm$^{b_g-1}$]	700	25000	9000	8870.40	8997.90
b_g [-]	1	1	1	1	1
S_t [mm]	0	150	0.43	42.54	79.66
Surface routing					
k_1 [h]	0.9	40	8	20.64	18.82
k_2 [h]	0.1	15	0.7	0.10	2.33
Groundwater storage routing					
k_b [h^{-1} mm^{1-m}]	0	5000	5.04	4528.30	4833.43
m [-]	3	3	3	3	3
Q modifications					
Q_c [m^3/s]	-0.3	0.03	0	-0.3	-0.25
t_d [h]	0	20	2	0	2

Table A.2: ΔmNSE between DA and OL for the different periods and observation operators. The largest improvement per period is in bold. Performance decreases are in red

Inverse observation operator	P1	P2	P3	PFull
LR full	$2.41\ 10^{-3}$	$2.64\ 10^{-3}$	$1.87\ 10^{-4}$	$1.74\ 10^{-3}$
LR full no forest	$2.36\ 10^{-3}$	$2.90\ 10^{-3}$	$1.91\ 10^{-3}$	$\mathbf{1.75\ 10^{-3}}$
LaR full $\tau = 30$	$5.37\ 10^{-4}$	$2.62\ 10^{-3}$	$\mathbf{8.07\ 10^{-4}}$	$9.41\ 10^{-4}$
Linear ϵ-SVR full	$\mathbf{2.62\ 10^{-3}}$	$\mathbf{3.13\ 10^{-3}}$	$-2.37\ 10^{-5}$	$1.21\ 10^{-3}$
GPR full	$1.06\ 10^{-3}$	$2.69\ 10^{-3}$	$-5.55\ 10^{-4}$	$5.60\ 10^{-4}$

Table A.3: Δ|FHV| (in %) between DA and OL for the different periods and observation operators. Higher bias after DA is indicated in red. The largest improvement per period is in bold.

Inverse observation operator	P1	P2	P3	PFull
LR full	$\mathbf{-0.25}$	-0.40	$-5.25\ 10^{-2}$	0.13
LR full no forest	-0.22	-0.40	$-7.14\ 10^{-2}$	0.10
LaR full $\tau = 30$	-0.12	$\mathbf{-0.51}$	$\mathbf{-0.13}$	$\mathbf{2.01\ 10^{-2}}$
Linear ϵ-SVR full	-0.21	$\mathbf{-0.51}$	$-3.59\ 10^{-2}$	0.12
GPR full	-0.14	-0.41	$-2.96\ 10^{-2}$	$7.21\ 10^{-2}$