Universiteit Leiden

Master Computer Science

AutoML for creating hybrid Earth science models

Name:	Victor Neuteboom
Student ID:	s1281437
Date:	23/11/2021
Specialisation:	Advanced Data Analytics
1st supervisor:	dr. Mitra Baratchi
2nd supervisor:	Prof. dr. ir. Peter van Bodegom

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Due to the availability of large sets of satellite data, an increasing number of Earth system science problems are tackled by applying machine learning. In general, two types of methods are used for Earth system science problems: "data-driven" methods and "theory-driven" methods. Data-driven methods involve the use of a large training dataset to train a machine learning model. In the context of remote sensing tasks, a machine learning model is trained by using a large set of "in situ" training data (ground truth measurements) coupled with satellite observations, where the satellite observations provide the input features and the in situ training dataset contains the target values to predict. However, in many scenarios the amount of available in situ data is limited. Theory-driven methods rely on the use of existing domain knowledge instead of large sets of training data. An example of such a method is the use of simulation models to create simulated training data. On the downside, these models typically require extensive domain knowledge to tune correctly.

A novel perspective on data science aims to combine these data-driven and theory-driven methods: "theory-guided" data science. In this thesis, we introduce a theory-guided framework that incorporates both simulation models and available in situ data within a modelling pipeline. For this framework, we create an extension to the existing automated machine learning framework of Auto-sklearn. We compare the performance of this new framework to several commonly used data-driven baselines including Random forest, Multilayer perceptron, Gaussian process regression and vanilla Auto-sklearn. To facilitate this comparison, we introduce a benchmark dataset consisting of four distinct Earth system science tasks with preprocessed, ready-to-use in situ, simulation and remote sensing data for each task. From our experiments with this benchmark dataset, we conclude that for one task (leaf area index estimation), the theory-guided framework outperforms all baselines. In this task, the proposed method improves on vanilla Auto-sklearn by an increase in R^2 of 0.01 to 0.02 for training sizes of up to 250 in situ samples. For other tasks, vanilla Auto-sklearn consistently ranks as the best model.

Contents

1 Introduction

Increasing availability of very large datasets have lead to great advancements in the application of machine learning techniques to typical computer science problems, ranging from computer vision to natural language processing. For example, the introduction of the ImageNet [16] dataset consisting of over 14 million manually annotated images and over 20,000 different categories, has allowed researchers to build and train very complex deep learning models that can achieve state-of-the-art results on a variety of computer vision tasks by leveraging information learned from ImageNet to different tasks [63, 85, 27].

In the Earth system science domain, we can find very similar tasks to those found in typical computer science problems. In a recent publication on the application of deep learning for Earth system science, Reichstein et al. [78] have identified several Earth system science problems that are very similar to well-researched computer science problems. For instance, the task of recognizing and segmenting objects within a photograph can be compared to recognizing extreme weather patterns in satellite images. Another example is video prediction, where we deal with multi-dimensional data that changes over time and try to predict a future state [78]. Such dynamic and multi-dimensional problems are abundant in Earth system science, ranging from tasks as vegetation modelling [81] to sea temperature forecasting [101]. In Figure 1, we have visualized these equivalencies between computer science and Earth system science tasks.

Despite these similarities, it has proven to be difficult to reproduce the results from these computer science tasks in their Earth system science counterparts [78]. This discrepancy can be explained by some key differences between the two domains.

Firstly, Earth system science problems can have a very high dimensionality. Where typical image recognition tasks work with three bands (red, green and blue), hyperspectral images used in Earth system science can cover hundreds of spectral bands. Furthermore, the image sizes vary hugely. Where typically images of 512 by 512 pixels are used in computer science, satellite data can include images of approximately 40,000 by 20,000 pixels [78].

Secondly, huge datasets do exist within the Earth system science domain (e.g., large collections of satellite data such those hosted on Google Earth Engine [28]), but they typically do not include huge numbers of labeled ground truth data. Usually, it is hard or expensive to gather ground truth data as it commonly requires physically sampling an object or area at a specific location and involves quite some manual labor. This type of ground truth data obtained from sampling an object or area is often referred to as "in situ" data.

Thirdly, even in situations where enough labelled ground truth (or in situ) data is available, there is a high risk of training a machine learning model that generalizes poorly. Frequently, Earth system science problems have complex underlying physical processes which are hard to capture by machine learning models. This may lead to overfitting models on the

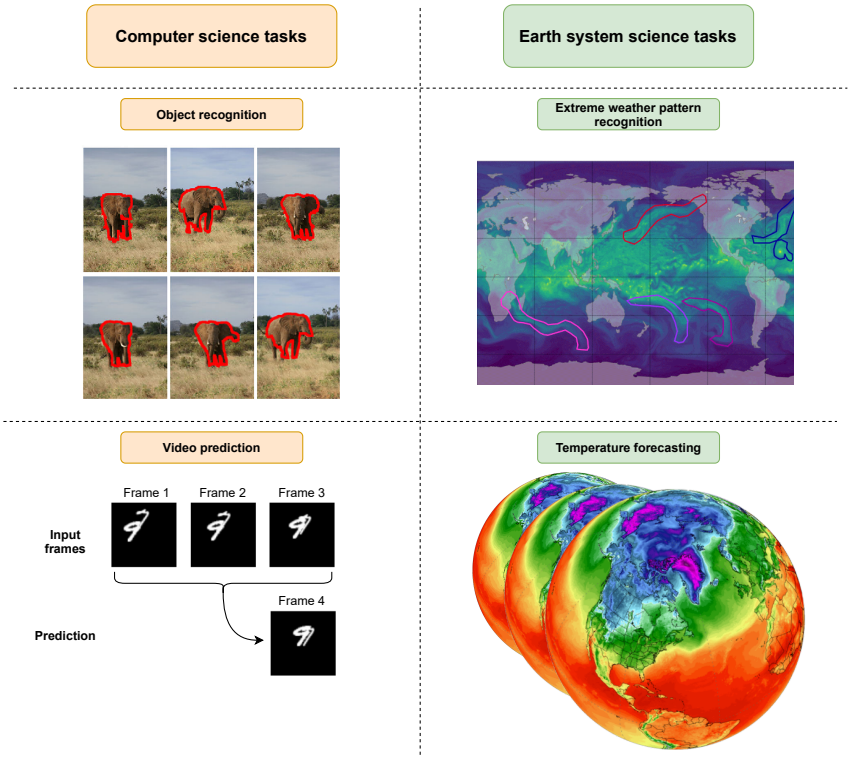


Figure 1: Examples of typical computer science tasks and their Earth system science equivalents (adapted from Reichstein et al. [78], Guillaumin et al. [29], Kashinath et al. [51], Joshi [48] and Climate Reanalyzer [47]).

available training data and thereby limiting the model's generalization capabilities [43].

In order to capture the underlying physical processes of these problems, physical simulation models are often used. For example, in vegetation monitoring tasks, radiative transfer models (RTMs) are used. These radiative transfer models simulate the surface reflectance of vegetation, similarly to what is measured by satellites such as Sentinel-2 [7]. In order to create these surface reflectance simulations, the radiative transfer model requires some input parameters to be set. For vegetation monitoring, these typically are (bio)physical parameters such as leaf area [70] and chlorophyll content [69]. The use of such simulation models allows researchers to tackle Earth system science tasks without the use of huge sets of labelled data. Moreover, as the simulation model is based on known physical processes, its results can be more physically consistent than that of a machine learning approach. However, in practice it is quite hard to parameterize these simulation models, requiring a great deal of domain knowledge. Furthermore, for a lot of problems, the underlying physical processes are not fully understood and as a result are hard to capture in a simulation model.

We have now discussed two approaches to Earth system science tasks: either use a model that learns from a large set of labelled ground truth (in situ) data, or use a (simulation)

model that is based on known theory. In the rest of this work, we will refer to this first approach as a "data-driven" approach as it fully relies on the use of a (large) dataset. We will refer to the second approach as a "theory-driven" approach, as it relies on the use of a (simulation) model that is based on known physical processes and/or other forms of domain knowledge.

Recently, a new perspective on data science was introduced which aims to combine the data-driven approach with a theory-driven approach: theory-guided data science [50]. The idea of theory-guided data science is to leverage the strong points of both approaches while negating their weak points. For instance, data-driven methods can learn to capture very complex or currently unknown physical processes and relations purely from data, whereas theory-driven methods can only be based on currently understood processes or theory. Furthermore, where data-driven methods can lead to physically inconsistent results, they can leverage theory-driven methods to correct their predictions and enforce physical consistency. This concept of combining data and theory is sometimes also referred to as hybrid modelling or physics-informed modelling. In the rest of this thesis, we will refer to approaches combining both data-driven and theory-driven aspects as "theory-guided". Theory-guided methods offer great potential for problems where a theory-based component is present, such as a physics-based simulation model or some known differential equations [78]. Unfortunately, there are some challenges for applying theory-guided methods. First of all, current works on theory-guided methods often focus on one specific problem domain and are not directly applicable to other problem domains. Without extensive domain knowledge, it can be very hard to adapt these methods to work in other problem domains. Secondly, due to having both a theory-driven and a data-driven component, the amount of design choices to make is heavily increased as both components will most likely have some design decisions to be made or (hyper)parameters to be tuned.

The goal of this thesis is to introduce a framework to automatically combine simulation models and data-driven models for tasks in the Earth system science domain. Within this goal, we can identify three distinct research goals.

Firstly, we aim to combine simulation models with data-driven models. This fits within the scope of theory-guided data science, which is a relatively new research field that encompasses techniques that aim to use available domain knowledge as well as state-of-the-art machine learning techniques.

Secondly, we aim to combine these models in an automated fashion. Therefore, we look at existing methods within the automated machine learning (or AutoML) domain. Automated machine learning methods aim to optimize steps in a modeling pipeline, by for example performing model selection or tuning a model's hyperparameters.

Thirdly, we aim to apply the techniques from these first two research areas to tasks within Earth system science. Typically, these tasks have a limited amount of measured in situ data. Instead, they rely on existing domain knowledge and techniques to gather more

proxy data such as data generated by remote sensing. As a consequence, tasks within this domain are well suited for combining theory and data-driven techniques.

We address these research goals with the following main contributions of this thesis:

1. We propose a novel method to combine ensembles of different theory-driven models with a data-driven component. We base this method on the state-of-the-art AutoML framework of Auto-sklearn.
2. We propose a benchmark dataset consisting of four distinct Earth system science tasks with relevant preprocessed and ready-to-use satellite data, simulation models and in situ data, to evaluate our method.
3. We compare our proposed theory-guided method to fully theory-driven and fully data-driven approaches that are frequently used in existing works.

In the next chapter, we will provide some definitions of keywords and formalize the problem setting. In Chapter 3 we will discuss related work in Theory-guided data science and Automated machine learning and their applications in Earth system science. Chapter 4 will cover several theory-driven and data-driven models that are used in our experiments. In Chapter 5, we will introduce our proposed method. Chapter 6 will describe our benchmark dataset and experimental setup. Chapter 7 will cover the results of these experiments and in Chapter 8 we will draw conclusions and list possibilities for future research projects.

2 Problem statement

In this chapter, we will introduce relevant definitions and terminology for the topics of theory-guided data science, automated machine learning and Earth system science. Furthermore, we will formalize our problem setting in a problem definition.

2.1 Definitions

2.1.1 Theory-guided data science

Data-driven: Methods that infer relations strictly from data, no extra knowledge is added. Traditional machine learning methods fall within this category, as they learn purely from using a dataset.

Theory-driven: Methods that infer relations strictly from knowledge, no extra data is added. Examples of theory-driven models are simulation models that are based on known physical laws or interactions. It could be argued that some of these models are not purely theory-driven, as they often contain components that are (partially) based on empirical results. For the scope of this thesis, we will still regard these models as theory-driven.

Theory-guided: Methods that incorporate both data and knowledge. For example, methods that incorporate simulation models as well as real data can be considered theory-guided. See Figure 2 for a comparison of theory-driven, data-driven and theory-guided models.

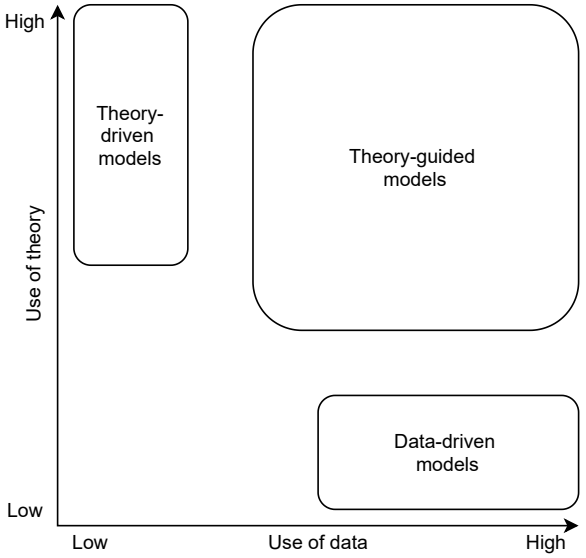


Figure 2: Relation between theory-driven, theory-guided and data-driven models (adapted from Karpatne et al. [50]).

2.1.2 Automated machine learning

Hyperparameter optimisation: Most machine learning models have hyperparameters that have to be tuned before training the model with data. Depending on the type of model, tuning hyperparameters can have a significant impact on model performance. The problem of finding an optimal set of hyperparameters for a model is called hyperparameter optimization.

Model selection: Besides tuning hyperparameters, an important decision when designing machine learning pipelines is the choice of machine learning model. This choice in turn affects the search space, as which hyperparameters can be tuned depends on the type of model. The problem of model selection and hyperparameter optimization is often combined and described as a Combined Algorithm Selection and Hyperparameter optimization (CASH) problem [92].

2.1.3 Earth system science

Remote sensing: With remote sensing, it is possible to gather data about an object from a distance, without physical sampling. In the context of Earth system science, remote sensing is typically done using satellites orbiting the Earth. These satellites measure reflectance in different wavelength bands. Remote sensing data describes reflectance values for a particular geographic location at a particular point in time.

Simulation data: A simulation model is a digital representation of a real world process and is often based on known physical laws or processes. With such a simulation model, it is possible to generate new data. The ease of generating simulation data depends on the complexity of the physical problem and the simulation software that represents it. A concrete example of such a simulation model for Earth system science tasks can be found in Chapter 4.1.

In situ data: Data can also be gathered by physically sampling an object or area, this is commonly referred to as in situ data. Sometimes in situ measurements are destructive, for example biomass is measured by harvesting a plant, destroying it in the process. Typically, in situ data is scarce as it is difficult or expensive to gather. In situ data can be used to validate or tune simulation models and remote sensing products.

2.2 Problem definition

In this thesis, we aim to develop a method that can automatically create a modelling pipeline from theory-driven and data-driven models. Practically, this is a Combined Algorithm Selection Problem (CASH) [41], where the search space is determined by the choice of theory-driven model, the choice of data-driven model and their respective hyperparameters. For a purely data-driven model, this problem can be formalized as follows:

$$\lambda^* = \underset{\lambda \in \Lambda}{\operatorname{argmin}} \mathcal{L}(\mathcal{A}_\lambda, D_{train}, D_{validation}), \tag{1}$$

where $\mathcal{L}(\mathcal{A}_\lambda, D_{train}, D_{validation})$ denotes a loss function that measures the performance of algorithm ($\mathcal{A}$ with the hyperparameter configuration λ. Here, algorithm $\mathcal{A}$ is trained on a training dataset D_{train} and validated on the validation set $D_{validation}$. The search space Λ can be represented as a tree, where the root node consists of the choice of model and all other nodes describe hyperparameters.

If we also incorporate simulation data into the pipeline, we could formalize the problem as follows:

$$\lambda^* = \underset{\lambda \in \Lambda}{\operatorname{argmin}} \mathcal{L}(\mathcal{A}_\lambda, D_{train\_in\_situ}, D_{simulation}, D_{validation\_in\_situ}), \tag{2}$$

where $\mathcal{L}(\mathcal{A}_\lambda, D_{train\_in\_situ}, D_{simulation}, D_{validation\_in\_situ})$ denotes a loss function that measures the performance of algorithm ($\mathcal{A}$ with the hyperparameter configuration λ. Here, algorithm $\mathcal{A}$ is trained on a training dataset consisting of in situ training data $D_{train\_in\_situ}$ and simulation training data $D_{simulation}$. It is then validated on a validation set $D_{validation\_in\_situ}$ consisting of only in situ data.

3 Related work

In this chapter, we will cover current literature relating to Theory-guided data science, Automated machine learning and their applications to Earth system science tasks. Furthermore, we will describe how this thesis relates to the literature.

3.1 Theory-guided data science

The field of theory-guided data science (TGDS) is still in its infancy. One of the first papers describing this perspective on data science was written by Karpatne et al. [50] in 2017. In this paper, theory-guided data science is defined as "an emerging paradigm that aims to leverage the wealth of scientific knowledge for improving the effectiveness of data science models in enabling scientific discovery" [50].

The emergence of this new perspective is motivated by the strengths and drawbacks of both purely theory-based and purely data-driven models. Data-driven models are able to learn complex and abstract relationships from large datasets. On the other hand, they are held back by their black-box nature, susceptibility to generalization errors (especially when dealing with limited training data) and the fact that they are often unsuitable for discovery of new knowledge [50]. Theory-driven models on the other hand are constrained to already known physical principles and theories thereby limiting their performance in problems with (partially) unknown dynamics. Theory-guided data science aims to combine the best of both worlds and in this paper several research themes and approaches are introduced along with practical examples. In this work, theory-guided data science techniques are grouped into several categories. We will describe these categories and give some practical examples for each category in the following subsections.

3.1.1 Theory-guided design

Model design involves the choice of model family (e.g., tree-based models) as well as any design choices within that model family (e.g., neural network architecture). The purpose of theory-guided design is to attain physically consistent results by basing the design of a data-driven model on available domain knowledge.

Neural networks offer a great deal flexibility in their design space and are therefore used in several papers that explore theory-guided design solutions. An example of such a work is a case study on electrochemical micro-machining, where a customized hidden layer was used [60]. In this custom hidden layer, several known physical relations were encoded in predefined connections and weights. In this case study, they found that the theory-guided neural network architecture outperformed purely data-driven architectures.

3.1.2 Theory-guided learning

In data-driven models, optimization algorithms are used to navigate a search space and to find an optimal solution to a specific problem. Theory-guided learning focuses on designing the learning algorithm based on domain knowledge or theory.

One aspect of theory-guided learning aims at warm-starting the learning process through the use of theory. A well-known way of warm-starting the learning process in neural networks is transfer learning. With transfer learning, a neural network model is pre-trained on a large dataset that is related to the problem dataset. For example, image recognition networks are typically pre-trained on the ImageNet [16] database. For problems where there is a small amount of available real world data, transfer learning can be used to pre-train a model on a dataset created by a simulation model and then finetuned on the real world data. This use of simulation data for pre-training was adopted in research regarding lake temperature modelling [46]. In this research, the authors found that pre-training reduced the need of having real world data. Even when only using a small subset of their real world data for finetuning, the pre-trained models outperformed other baselines.

A second way of guiding the learning algorithm through the use of theory is by the use of regularization. An example of using regularization is found in the work of Sahli Costabal et al. [83]. In their work, they added a regularization term based on the Eikonal equation to the loss function of a neural network that was trained to predict the electrical activation map of the heart. By the addition of the physics-based regularization term, the predictions of the neural network were guided to be more physically consistent and achieved increased robustness and consistency when dealing with different sets of training samples.

Finally, the learning process can be guided by the use of constraints. These constraints can be explicitly defined, for example in a partial differential equation (PDE), or implicitly derived. Explicit constraints from partial differential equations are used in Physics-Informed Neural Networks [72]. These networks consist of two parts. The first part encodes input data into latent variables. The second part uses these latent variables along with the partial differential equation to obtain physically consistent results.

However, such explicit constraints are not available for every problem. For some of these problems, explicit constraints may not be known, but implicit constraints may be built into a simulation model. These implicit constraints could potentially be leveraged by data-driven models. In 2018 Ren et al. investigated whether they could capture these constraints from a simulation model by using a generative adversarial neural network (GAN) [79]. From their experiments, they concluded that the (semi-supervised) adversarial networks were able to outperform other supervised learning models.

3.1.3 Theory-guided refinement

Data-driven models can be powerful tools in modelling physical problems. However, the output they produce may not be in accordance with known physical laws or relations. With theory-guided refinement, the idea is to use these known physical laws or relations to refine the outputs of a data-driven model as a post-processing step. This technique was used in a research project where the aim was to discover new chemical compounds [31]. Potential new compounds were generated by using a machine learning model that was trained on a dataset consisting of existing compounds. Newly generated compounds were then checked with ab initio computations which involve known physical constants. Generated compounds that did not comply with these computations were pruned.

3.1.4 Data-driven augmentation

Where the previous categories of theory-guided techniques aimed to improve data-driven models by using domain knowledge, it is also possible to improve theory-based model by using data-driven techniques.

Theory-based simulation models usually have many parameters that need to be tuned. As these parameters are often of mixed types and influence each other, they form large and complex search spaces. Typically, these parameters describe (bio)physical values. For example, for a simulation model that estimates lake temperature uses physical parameters such as incoming radiation, water clarity and wind speed to guide the simulation of water temperature at several depths of the lake [49]. Here incoming radiation, water clarity and wind speed are used as input parameters for the simulation model and water temperature is the derived simulation output. Data-driven techniques may be used to efficiently explore and exploit these search spaces and to find suitable simulation configurations. This technique was applied to stochastic radio propagation models which were calibrated using a Multilayer perceptron (MLP) [1].

Instead of addressing the simulation model input parameters, it is also possible to refine its output with data-driven techniques. This is accomplished by using the simulation parameters and simulation output as input for a data-driven model. In this case, the data-driven model learns to correct the simulation output based on available real world data as visualized in Figure 3. Using the example of lake temperature modelling, the physical parameters (incoming radiation, water clarity and wind speed) are used as input for a lake simulation model, which outputs an estimated water temperature value. Then both the physical parameters and the estimated water temperature value are used as features in a machine learning model which makes a final estimation of the water temperature. Such an approach was used in research on warm forming processes [52] and lake temperature modelling [49].

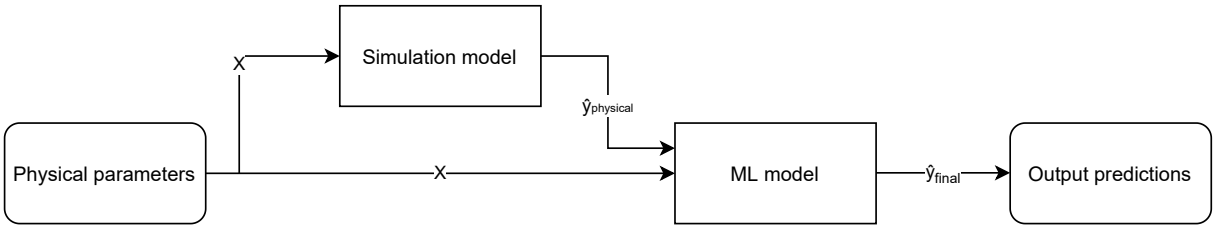


Figure 3: Data-driven augmentation (adapted from Karpatne et al. [49])

3.1.5 Application to Earth system science

In 2019, the idea of combining theory and data-driven models was described specifically for Earth system science [78]. The huge increase of available data (mainly from remote sensing) has allowed the application of deep learning techniques to geoscientific problems such as land cover classification and soil mapping. Similarly to the findings of Karpatne et al. [50], important caveats of machine learning techniques are noted for application in the geoscience domain. These pitfalls include data biases and extrapolation errors. For the successful use of machine learning and deep learning specifically for Earth system science problems, the authors identify five important challenges: interpretability, physical consistency, data complexity and uncertainty, limited labels and computational demand.

They argue that these challenges can be tackled by the integration of physical and machine learning models and propose several points of synergy for the application of combined models.

Building further upon this work, in 2020 Camps-Valls et al. wrote their paper "Living in the Physics and Machine Learning Interplay for Earth Observation" [12]. In this work, they further elaborate on the options available for the synergy of theory-driven and data-driven models. Furthermore, they provide examples of practical use cases, ranging from merging data and simulations through Joint Gaussian Processes to learning ordinary differential equations (ODE) with sparse regression.

3.1.6 Summary & relation to this thesis

One of the goals of this thesis is to explore the possibility of introducing theory-driven knowledge, for example encoded into simulation models, into data-driven modelling pipelines for Earth system science problems. This idea fits into the perspective of theory-guided data science, which aims to combine theory and data in model design, learning processes or in model refinement.

Theory-guided data science offers an opportunity for fields where there is some data available to apply machine learning methods, but not enough to support state-of-the-art methods such as complex deep learning models. A prerequisite for theory-guided data science is that some aspect of the problem should be theory-driven. An example of a

theory-driven aspect would be encoding known physical laws, such as conservation of energy, into a physical simulation model.

Earth system science has many problems which fit within these criteria: the amount of real "in situ" data is often limited and current methods often rely on theory for modelling problems. As such, these problems are good candidates to investigate the application of theory-guided techniques.

In this thesis, we explore the idea of data-driven augmentation described in Section 3.1.4 by incorporating this technique into our proposed method. Current research considering theory-guided data science techniques usually focus on one particular problem or dataset. However, in this thesis, we apply this proposed method to several different Earth system science problems in our introduced benchmark dataset and compare its performance to baselines including frequently used machine learning methods.

3.2 Automated machine learning

In the process of building a machine learning modelling pipeline, many design choices have to be made. For example, one needs to determine which type of modelling technique to use and which hyperparameters to use to lead to the best results. Furthermore, feature and data preprocessing steps need to be considered. Combining all these possible design choices results in a search space that is simply too large to fully explore. In this section, we will look at current research in automating these design choices with automated machine learning (or AutoML) systems.

3.2.1 Model selection & hyperparameter optimization

The need for model selection and hyperparameter optimization is based on two important problems. Firstly, there is no single model that performs best on all possible datasets. Secondly, some types of models rely heavily on a correct choice of hyperparameters to perform well [41].

The problem of selecting a model and optimizing hyperparameters is known in literature as Combined Algorithm Selection and Hyperparameter optimization (or CASH). A common way of tacking this problem is by the use of Bayesian optimization. The main idea of using Bayesian optimization for CASH problems is to create a probabilistic model that learns to represent a relationship between model choice and hyperparameters on one side and model performance on the other side [41].

This probabilistic model can be created using Gaussian process regression, where the posterior distribution induced by the Gaussian process model can be used to decide which parameters to try in a next iteration [87]. Another approach to the CASH problem is the use of tree-based models. An example of a tree-based approach is Sequential Model-based Algorithm Configuration (or SMAC), which is based on random forests [40]. Where Gaussian process models seem to perform better for problems with purely numerical

parameters and low dimensionality, they seem to be outperformed by tree-based models for high-dimensional problems, especially in structured and semi-discrete problem settings [41, 21].

Automated configurators can be used to build automated machine learning systems that make it easy for users to create a modelling pipeline with optimal hyperparameters and model selection. An example of such a system is AutoWEKA [92]. AutoWEKA is based on the WEKA [30], a data mining software package that includes implementations of many different estimators. AutoWEKA applies SMAC to automatically choose an estimator from the WEKA suite and to find an optimal set of hyperparameters.

3.2.2 Meta learning

Automated configurators can be used to generate modelling pipelines that are on-par or better than their humanly designed counterparts. However, the optimization process is expensive: for each iteration a machine learning model with a particular set of hyperparameters needs to be trained, which can be computationally expensive. The process of finding an optimal model can be sped up by the use of meta learning [41]. The key concept behind meta learning is to learn from previous experiences of finding optimal models. To do so, first a set of meta data is collected which describes previous optimization tasks and the resulting models. This set of meta data contains features that describe the resulting model, as well as features that describe properties of the original dataset. Meta learning can be used to instantiate the search process for a CASH problem. For example, in the AutoML system Auto-sklearn meta learning is used to warm-start the search process with model configurations that worked well for similar datasets (where the similarity is based on nearest neighbors in meta-feature space) [23].

3.2.3 Summary & relation to this thesis

Using automated machine learning techniques, it is possible to automatically create machine learning pipelines whose performance rival that of manually engineered pipelines. The problem of selecting a model and its hyperparameters can be tackled using Bayesian optimization. Even though this is a computationally expensive process, there are some tricks to speed it up. For example, the use of meta learning can warm-start the search process for an optimal model. In this thesis, we intend to use automated machine learning techniques to create a modelling pipeline that includes both a theory-driven and a data-driven aspect. To achieve this, several steps of model selection and hyperparameter tuning need to be applied. Therefore, any automated machine learning techniques that tackle CASH problems can be useful in this context.

In this thesis, we use the existing AutoML framework auto-sklearn [23] as the basis for our proposed method. In this method, we use auto-sklearn to automatically combine simulation models with machine learning models and to tune their hyperparameters. Auto-sklearn was chosen as it supports machine learning methods that are frequently

used in Earth system science problems (such as Random forest, Multilayer perceptron and Gaussian process regression) and can be easily extended with customized components [24].

4 Background

In our experiments, we use a variety of theory-driven and data-driven models. In this chapter, we will briefly describe how these models work and how they can be used in a modelling pipeline.

4.1 Theory-driven models

Any model that is purely based on theory, such as known physical laws, can be considered a theory-driven model. In the scope of this thesis, we focus on physical simulation models. As the Earth system tasks that we experiment with all involve remote sensing, we focus on radiative transfer models and how to inverse them. For more information on these tasks, see Section 6.1.

4.1.1 Radiative transfer models

For remote sensing applications, radiative transfer models (RTMs) are used to simulate reflectance spectra based on a set of physical parameters. These models are based on a radiative transfer equation. This equation describes how energy is transferred when radiation travels through a medium [14]. It deals with the physical processes of absorption, refraction and scattering.

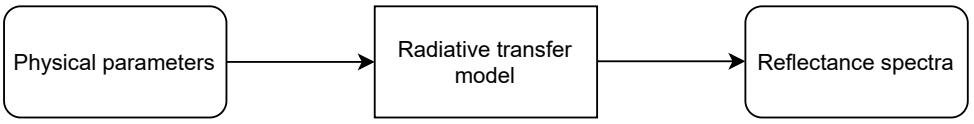


Figure 4: Generating reflectance spectra based on physical parameters using a radiative transfer model.

An example of an often used radiative transfer model is PROSAIL [44]. PROSAIL is used to model light interactions in plant canopies based on biophysical parameters. It is composed of two physical submodels, PROSPECT and SAIL where PROSPECT models light interactions at the leaf level and SAIL models interactions at the canopy level. The biophysical parameters that can be tuned for PROSPECT and SAIL are given in Tables 1 and 2. Based on these biophysical parameters PROSAIL simulates surface reflectance values for different wavelengths, as visualized in Figure 4.

Parameter	Symbol	Unit
Chlorophyll content	C_{ab}	μg cm^{-2}
Carotenoids content	C_{ar}	μg cm^{-2}
Brown pigment content	C_{bp}	Arbitrary units
Water equivalent thickness	C_w	cm
Dry matter content	C_m	g cm^{-2}
Leaf structure index	N	Arbitrary units

Table 1: PROSPECT parameters

Parameter	Symbol	Unit
Leaf area index	LAI	m^2 m^{-2}
Average leaf inclination angle	ALIA	Degrees
Hot spot size parameter	Hspot	Arbitrary units
Soil reflectance	p_{soil}	Arbitrary units
Soil brightness parameter	P_{soil}	Arbitrary units
Sun zenith angle	SZA	Degrees
Observer zenith angle	OZA	Degrees
Relative azimuth angle	RAA	Degrees
Ratio of diffuse to total incident radiation	skyl	Arbitrary units

Table 2: SAIL parameters

4.1.2 Inversion

A common strategy to incorporate simulation models, such the radiative transfer models described in previous section, into a modelling pipeline is by model inversion. The first step of the inversion procedure involves the creation of a lookup table [80]. This lookup table is created by performing many simulations with different input parameters. For example, you could create many simulations where you vary the leaf area index input parameter.

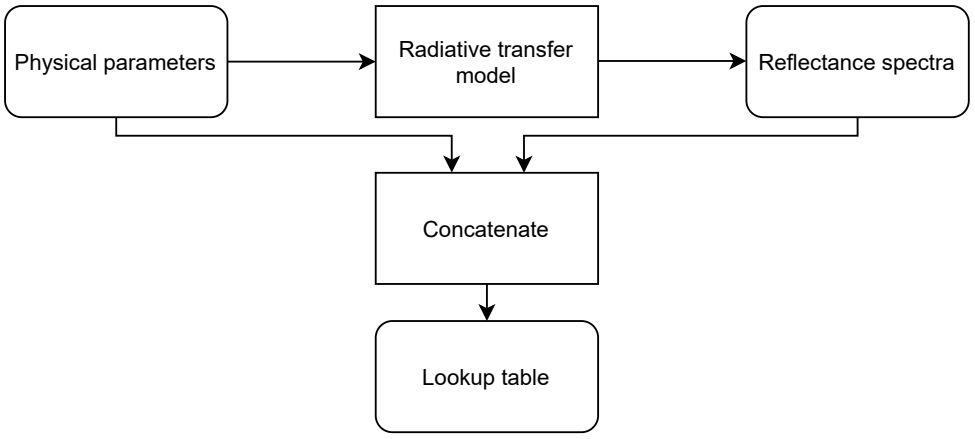


Figure 5: Generating a lookup table of reflectance data and physical parameters using a radiative transfer model.

After creating this lookup table, it could then used as a training set for a machine learning algorithm, where the leaf area index would be the target and the simulated reflectances the input features. In this way, the machine learning algorithm can learn the relationship between the simulated reflectance and the target variable (leaf area index). After learning this relationship, the machine learning model can now predict the target variable based on remote sensing reflectance values. This process is visualized in Figure 6. This process is called (lookup table) inversion and its incorporation into a modelling pipeline is often referred to as hybrid modelling.

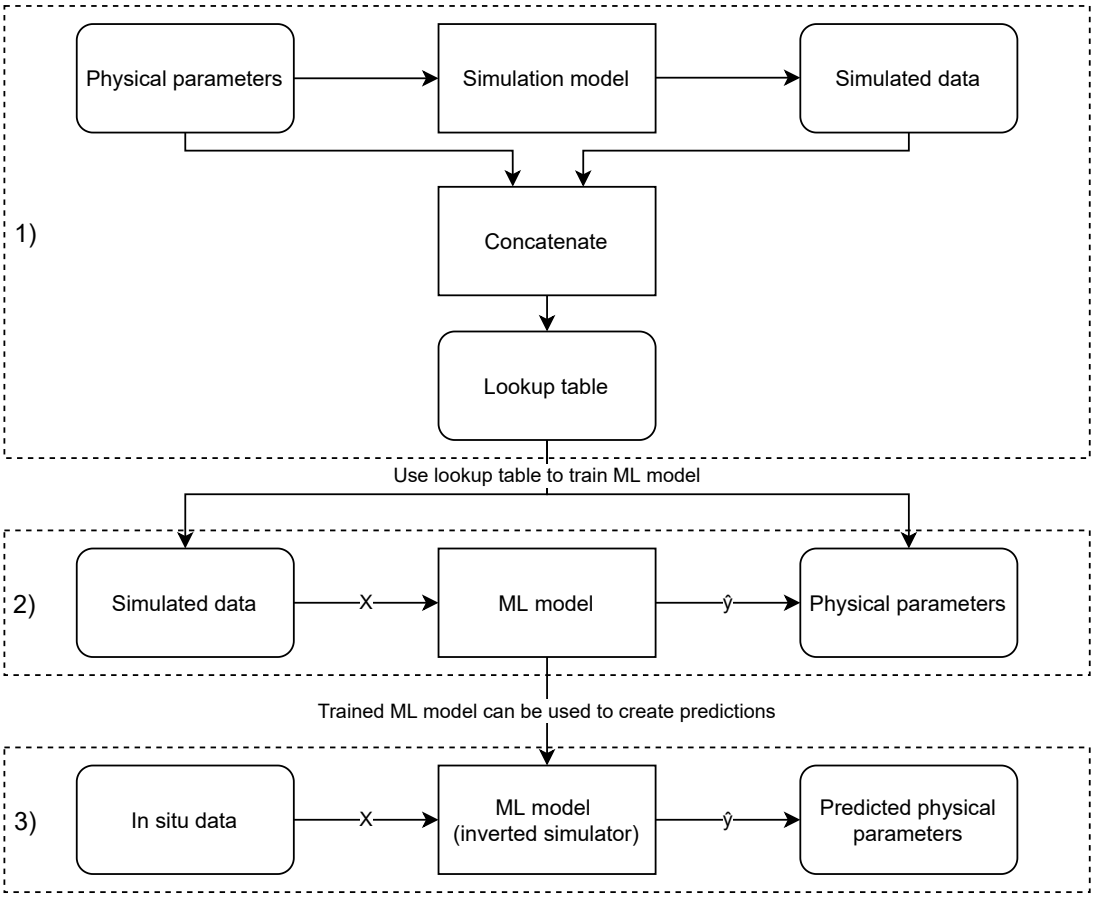


Figure 6: The inversion process in three steps. Step 1: create a lookup table. Step 2: train machine learning model on lookup table. Step 3: make predictions on real data with trained model.

This inversion process is unfortunately not without its challenges. One of the most challenging aspects of this process is that the problem is ill-posed [108]. For many different input parameter combinations, the simulation model can generate very similar reflectance curves. As a result, the achieved performance when testing this hybrid modelling approach on in situ data accompanied by remote sensing data is often limited. This ill-posedness creates a challenge for the application of data science techniques and inspires researchers to tackle the problem with novel methods.

4.2 Data-driven models

In our experiments, we will compare several data-driven models to a novel theory-guided model. Here, we chose commonly used machine learning methods as the data-driven models that will serve as a baseline.

In this Section we will briefly describe the data-driven models that are used in the

experiments. These range from relatively simple baseline models such as a random forest to more complex models such as the ensembles created by Auto-sklearn.

4.2.1 Machine learning

For our experiments, we selected a few modelling techniques as simple baselines. These modelling methods were selected as they are frequently used in Earth system science research problems that involve a machine learning component. Even though these models are relatively simple to implement, they have been shown to achieve good results. For example, a random forest model was used to estimate leaf area index retrieval with relatively high accuracy [57].

Random forest
The first modelling technique we chose as a simple baseline is the random forest model. In a 2016 review on the use of random forest for remote sensing problems the authors concluded that the random forest is capable of handling high dimensional data [5]. Furthermore, it was concluded that the random forest is quick to train and insensitive to overfitting.

Random forest is an ensemble method that builds an ensemble (or forest) of decision trees in parallel. Each decision tree creates a prediction and all predictions are averaged in order to obtain a final prediction from the random forest model. The trees are created using bootstrap aggregation (or bagging) [10]. In short, this means that for every tree a sub-sample of the training dataset is taken. This sub-sample is created by sampling with replacement. The random forest has several hyperparameters that can be tuned, these are described in Table 3.

Parameter	Description
Number of estimators	Number of trees in the random forest.
Criterion	Which metric is used to measure the quality of a split.
Max features	How many features can be considered when deciding which split is optimal.
Max depth	Maximum depth of each tree.
Min samples split	How many samples are needed to split a node.
Min weight fraction leaf	Lower limit of sum of weights needed to create a leaf node.
Bootstrap	Whether bootstrapping is used to build new trees.
Max leaf nodes	How many leaf nodes are created at most.
Min impurity decrease	Nodes can only be split if splitting results in a reduction of impurity of at least this parameter.

Table 3: Tuneable hyperparameters for random forest.

Gaussian process
Our second simple baseline is Gaussian process regression. We chose this baseline as

it is often used in plant parameter retrieval, which is a setting that is close to our experiments [11, 98]. Gaussian process regression is a non-parametric Bayesian approach [84]. Some of the advantages of using Gaussian process regression include the ability to create confidence intervals and allowing the use of active learning techniques. For Gaussian process regression we can tune the hyperparameters given in Table 4.

Parameter	Description
Kernel	Which kernel is used as covariance function.
Alpha	How much noise is added to training samples.
Optimizer	Which optimization function is used.
Number of optimizer restarts	How often the optimizer can be restarted to find optimal kernel parameters.
Normalize y	Whether target values are normalized or not.

Table 4: Tuneable hyperparameters for Gaussian process regression.

Multilayer perceptron

The Multilayer perceptron (MLP) is a type of feed-forward artificial neural network. With recent advances in deep learning, neural networks have become a popular tool for many data-driven problems. Multilayer perceptrons can learn to represent complex relationships and have been used with success in for example atmospheric science problems [25]. Neural networks are highly customizable, design choices have to be made regarding the shape of the hidden layers, the activation functions, how to optimize the weights and more [73].

To simplify the search space of Multilayer perceptron design choices, one can use templates with predefined shapes of hidden layers. With such a template, there is no need to tune the number of neurons for each hidden layer individually. Instead, the number of neurons per hidden layer depend on the number of neurons in the previous layer, resulting in requiring to tune only a single layer. This idea of shaped networks was used in the AutoML framework Auto-pytorch [107], based on shapes defined the autonomio Python package [54]. The Multilayer perceptron implementation used in our experiments utilizes a brick shape, meaning that each hidden layer has the same amount of neurons. All tuneable hyperparameters of this implementation are given in Table 5.

Parameter	Description
Hidden layer sizes	How many neurons are used in each hidden layer of the network.
Activation function	Which activation function is used for each hidden layer.
Solver	Which optimization function is used to determine weights.
Alpha	How much L2 regularization is applied.
Batch size	How many samples are used per batch during training.
Learning rate	Which schedule to use for learning rate.
Initial learning rate	Initial value for learning rate.

Table 5: Tuneable hyperparameters for Multilayer perceptron.

4.2.2 Automated machine learning

As a state-of-the-art baseline data-driven baseline, we use an automated machine learning system called Auto-sklearn [23]. The Auto-sklearn system is primarily based on three components: Bayesian optimization, meta learning and ensembling.

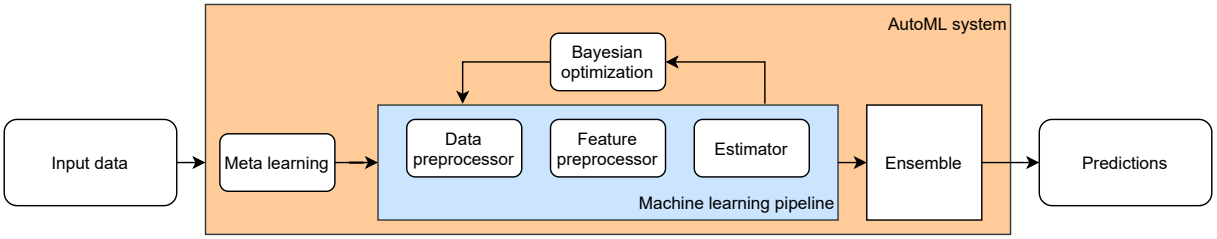


Figure 7: Auto-sklearn training and optimization approach (adapted from Feurer et al. [23])

The main component of Auto-sklearn consists of the Bayesian optimization process that is used to perform model selection and to tune hyperparameters. Auto-sklearn does not only choose which model type to use, but also automatically incorporates feature and data preprocessing steps into the modelling pipeline (see Figure 7).

The searching process for these modelling pipeline components is warm-started using meta learning. This meta learning approach is trained using a large number of datasets with according meta features and performance data. Then, for a new dataset, Auto-sklearn selects configurations to start the optimization process with based on a nearest-neighbors approach.

Finally, when Auto-sklearn has found several modelling pipelines that perform well, the system combines these pipelines in an ensemble. For regression problems, predictions are made in this ensemble by taking the mean of predictions made by individual pipelines in the ensemble. For classification problems, predictions are made using majority voting

[9]. The modelling pipelines to include in the final ensemble are chosen using ensemble selection. In short, ensemble selection is a greedy technique that iteratively adds the pipeline that minimizes the validation loss of the ensemble to the ensemble. This process continues until the maximum size of the ensemble is reached, or if the validation loss does not decrease when adding a new pipeline to the ensemble. It is possible to add the same pipeline to the ensemble twice.

5 Method

In contrast to the data-driven models introduced in last section, which only utilized in situ data, our proposed theory-guided model (or proposed method) uses a combination of simulation data and in situ data. This proposed method is primarily based on three components: stacked ensemble, data-driven augmentation and automated machine learning. Firstly, we will motivate our choice of techniques to base our method on. Secondly, we will briefly explain each of these techniques. Afterwards, we will discuss how these techniques are combined and implemented in the proposed theory-guided model.

5.1 Motivation

One of the main aims of this thesis is to introduce a framework to automatically combine simulation models and data-driven models. Due to the vast amount of existing data-driven, theory-driven and novel theory-guided methods, this is not a trivial task. In order to confine the scope of this task to an achievable level, we made some specific design choices.

Firstly, we chose to base our method on an existing AutoML framework: Auto-sklearn [23]. We chose to use Auto-sklearn for two main reasons: Auto-sklearn supports several machine learning methods that are widely used in Earth system science (e.g., Random forest, Multilayer perceptron and Gaussian process regression) and Auto-sklearn is customizable and can be extended with new components [24].

Secondly, we decided to focus on incorporating a specific theory-guided method as a way of combining theory-driven and data-driven models. Here, we chose to base our method on data-driven augmentation as proposed by Karpatne et al. [49]. This technique was chosen as it is very versatile. Where many theory-guided methods only work for very specific problems (e.g., Physics Informed Neural Networks require a partial differential equation to be known), data-driven augmentation can be applied to any problem that incorporates a simulation model and a data-driven component.

Thirdly, we chose to leverage the built-in ensembling functionality of Auto-sklearn in the data-driven augmentation component of the proposed method, creating a "stacked ensemble". The reasoning here is that the use of an ensemble of various theory-driven models may lead to improved generalization performance for the theory-driven part of the proposed method [55].

5.2 Data-driven augmentation

As explained in Section 3.1, one of the categories within theory-guided data science concerns data-driven augmentation. In short, the idea of data-driven augmentation is to use a data-driven model to correct (or augment) some aspect of a physical (or theory-based) model. This can be done in several ways. For example, you could calibrate a theory-

based (simulation) model by the use of machine learning [1] and thereby "augmenting" the input parameters of the theory-based model. Another option is to use a data-driven model to correct the output of a theory-based model [49]. In our case, we will focus on this second option. In this case, we assume we have a physical model that is given some input parameters (or features) and creates an output (or prediction) based on those input parameters. Instead of using the predictions made by the physical model directly as our final predictions, we feed them as input into a machine learning model along with the original input parameters. Then, this machine learning model can learn to correct for any biases or errors within the physical model. This scenario assumes that a physical model requires no additional training and that the machine learning model can be trained using available in situ data.

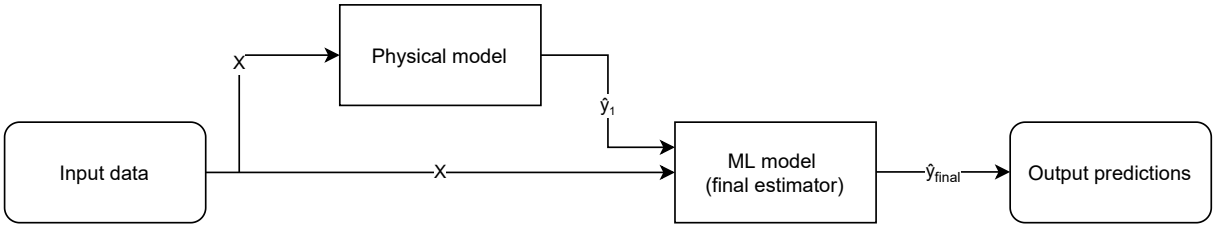


Figure 8: Data-driven augmentation: correcting the output of a physical model with machine learning.

For all tasks in our benchmark dataset, the physical model consists of an inverted simulation model as all tasks are considered inversion problems (the input of the non-inverted physical model is actually the desired target).

5.3 Stacked ensemble

The stacked ensemble [88] technique builds upon two separate concepts: ensembling and stacking. The main idea behind ensembling is that multiple models can be trained in parallel, where the outputs of each model can be combined by taking a (weighted) mean to obtain a final prediction. Creating such an ensemble of models can lead to a decrease in bias and therefore an increase in model performance when tested on unseen samples [88]. This decrease in bias can be attributed to the increased diversity of models in an ensemble. Due to this diversity of models, potential overfitting of one of the models has a reduced effect on the final prediction of the entire ensemble.

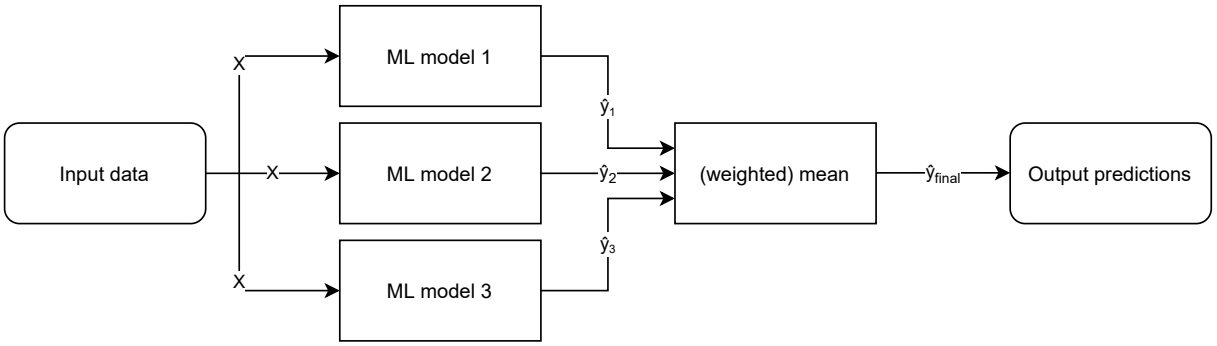


Figure 9: An ensemble of three parallel models.

A stacked ensemble is very similar to a regular ensemble, except for the way the predictions of the individual models are combined. Instead of simply taking a (weighted) mean of predictions, the predictions are used as input features to another, final, machine learning model. This final model learns how to combine the predictions generated by the models included in the ensemble. The entire pipeline consisting of the models in the ensemble as well as the final model can be optimized at the same time. The advantage of this method of ensembling is that the way the individual models are combined can be optimized, which can lead to increased model performance [66]. The final estimator can learn to favour the predictions of a particular model in the ensemble over the others based on their individual predictions. It could very well be that a particular model in the ensemble creates more accurate predictions within a certain range of target values. Training a stacked final estimator could allow to capture such a relationship, whereas simply taking a weighted average cannot.

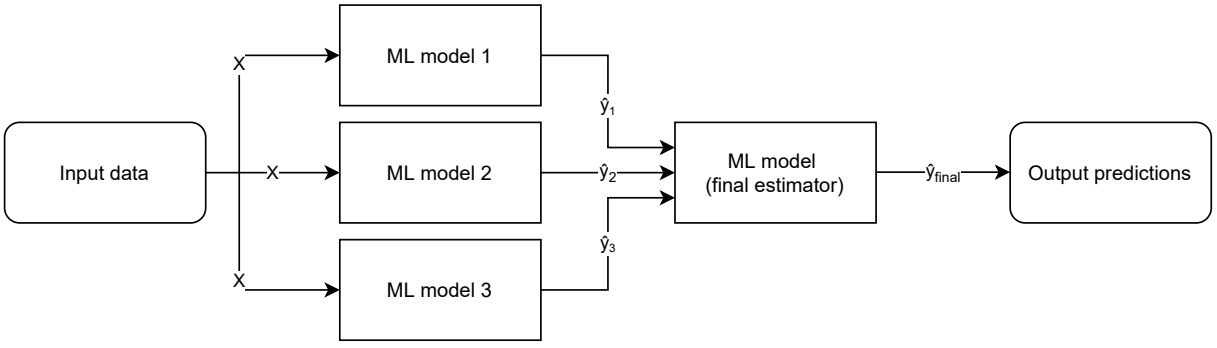


Figure 10: A stacked ensemble of three parallel models.

5.4 Proposed theory-guided framework

In our proposed framework, we bring the ideas of stacked ensembles and data-driven augmentation together. Furthermore, the architecture of the proposed method includes

a theory-driven and a data-driven part. Firstly, it consists of an ensemble of different physical models. In the context of our experiments, these physical models are all inverted simulators.

We incorporate the concept of data-driven augmentation by feeding the output of each inverted simulator, as well as the original input, into a machine learning model that is trained on only in situ data. This works in a similar way to the stacked ensemble technique, where the final model learns the proper way to combine the outputs of the models in the ensemble. However, as the original input features are also directly fed into the final model, it can also learn to correct the outputs from the models in the ensemble (data-driven augmentation). Put together, the final architecture is given in Figure 11.

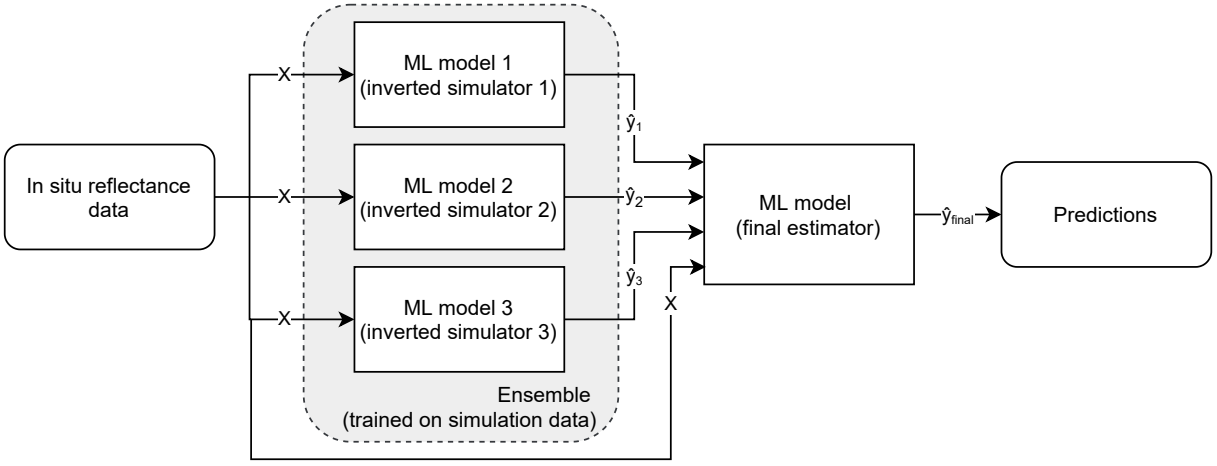


Figure 11: Architecture of proposed method.

The goal of our proposed method is to automatically create the a combination of data-driven and theory-driven models. In order to automatically create such a model, we need to define a search space that incorporates design choices for both the theory-driven and data-driven aspects of the model. To constrain the search space we base our implementation on the existing automated machine learning framework of Auto-sklearn.

Incorporating both a theory-driven part (ensemble of physical models) and a data-driven part (machine learning model) within one architecture is done by splitting our training procedure up into two steps. In the first step, we create and train the theory-driven part. As described in Section 6.1, for each task in our benchmark dataset we have a simulation lookup table created by a simulation model. Using this lookup table, we invert the simulation model by using data created by the simulator (e.g., reflectance spectra) as input data and a simulation parameter (e.g., leaf area index) as target data. With these input data and target data, we create and train an ensemble of models using Auto-sklearn. As explained in Section 4.2.2, Auto-sklearn tries to find an optimal combination of data preprocessing steps, feature preprocessing steps and estimator hyperparameters

using Bayesian optimization. The search for this optimal combination is instantiated through the use of meta learning. Each model of the resulting Auto-sklearn ensemble can be considered a unique inverted simulator.

Normally, Auto-sklearn combines the models from its ensemble using a weighted mean. Instead, we replace this weighted mean with another final machine learning model. This final machine model represents the data-driven part of the architecture. In the second step of our training procedure, we train and tune this model using available in situ data. For each sample in the in situ data, we use it as an input for each of the inverted simulators. Each inverted simulator then creates a prediction based on that input, resulting in a prediction for each inverted simulator in the ensemble. These predictions are concatenated to each other and to the original in situ input sample. Then, these concatenated predictions and input data are fed into the final machine learning estimator. Again, we use the Bayesian optimization and meta learning approach from Auto-sklearn to find an optimal estimator (including data and feature preprocessing steps and optimal hyperparameters). This estimator creates a final prediction. During training, this model will thus learn to combine and correct the outputs of several inverted simulators along with real input data.

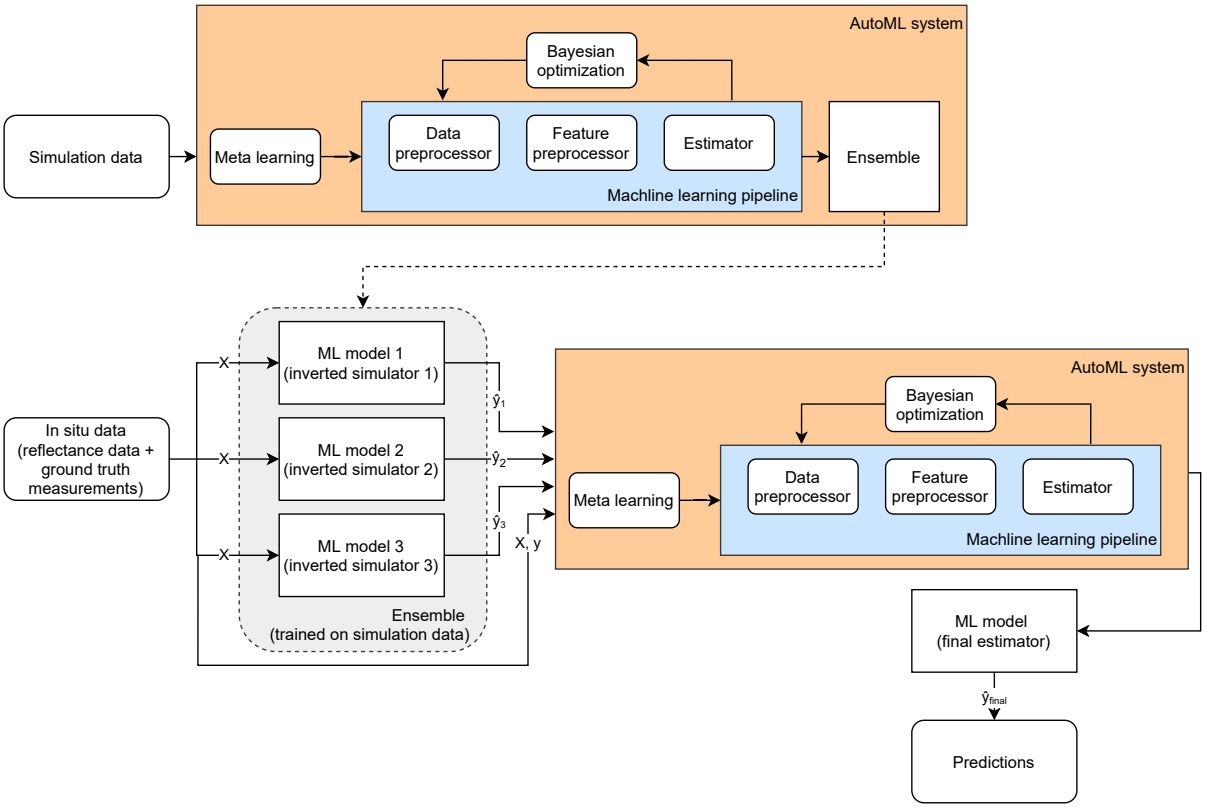


Figure 12: Training and optimization procedure of proposed method (partially adapted from Feurer et al. [23]). Step 1: create ensemble of inverted simulators (theory-driven). Step 2: train machine learning model as final estimator (data-driven).

As both the theory-driven part and the data-driven part of the model architecture are created by Auto-sklearn, the search space of both parts is determined by the used settings for Auto-sklearn. For example, the hyperparameter search space is determined by the selection of models that are included in the model search space. These settings are given for both training steps in Table 6.

Training step	Models	Ensemble size	Best models in ensemble	Initial meta-learning configurations
1 (theory-driven)	All*	5	5	5
2 (data-driven)	All*	1, 5	1	1

Table 6: Auto-sklearn parameters used in first step of proposed method training procedure.

*These include all models that are in the default configuration of Auto-sklearn.

6 Experiments

The purpose of the experiments in this thesis is threefold. Firstly, they serve as a validation exercise of the proposed benchmark dataset. Secondly, they explore the relationship between the amount of available training data and the performance of both data-driven and theory-guided techniques. Finally, the usefulness of automated configuration is tested.

In this chapter, we will first introduce the benchmark dataset and the Earth system science tasks contained within this dataset. Afterwards, we will explain which modelling approaches will be used in experiments. Furthermore, we will elaborate on used automated machine learning methods and the search space that was used in experiments. Finally, we will describe how the experiments combine the benchmark data, modelling approaches and automated machine learning methods in a structured way.

6.1 Benchmark data

In order to properly judge the performance of modelling techniques, whether or not they are purely data-driven or theory-guided, a dataset is needed to experiment with. Preferably, a quality controlled benchmark dataset should be used. Unfortunately, such benchmark datasets are scarce in the Earth system science domain. This scarcity was concluded by Ebert-Uphoff et al. in their 2017 paper [20] where they present a vision on the development of benchmark datasets for geoscience problems and call on the data science and earth system science research communities to present publicly available benchmark datasets. In their vision on benchmark datasets, they introduce several conditions a benchmark should preferably satisfy. Among others, these conditions include the following:

1. **High impact**: The task or problem associated with the dataset should have a clear contribution to either active research or should have clear societal benefits.
2. **Active research area**: Interaction and cooperation between earth system science and data science researchers should be stimulated. Therefore, the data should come from an active research area.
3. **Challenge for data science**: The data science task should not be too simple to perform. The benchmark should clearly specify an active research challenge. Otherwise, data science techniques may not be required for that specific task.
4. **Generality and versatility**: Knowledge gained by researching the dataset should ideally be transferable to similar Earth system science problems. Examples may include used modelling techniques.
5. **Evaluation**: To know whether applied data science techniques are successful, some measure of performance should be included in the task description of the benchmark. This also allows other researchers to compare their results to existing ones.

Some benchmark datasets for Earth system science problems already exist. An example

of such a benchmark is the WeatherBench dataset [75]. However, as the title of the accompanying paper "WeatherBench: a benchmark data set for data-driven weather forecasting" suggests, this dataset is primarily meant for purely data-driven methods. The dataset consists of a very large volume of available data, whereas in many other Earth system science problems only a very limited amount of in situ data is available. These limited data cases are where the largest gains can be made by using a combination of theory and data-driven techniques.

As a result, we opted to create our own benchmark set of data sources that focus on these low data availability scenarios. Besides concerning low data availability problems, we also only considered problems where some kind of simulation model is available to use as a source of domain knowledge. Moreover, each data source should be publicly available and preferably quality controlled in some way.

With these criteria in mind, we found four suitable tasks with accompanying datasets. These four tasks and datasets satisfy the five conditions from Ebert-Uphoff et al. [20] mentioned earlier. Each of these tasks are part of active research topics that closely relate to overarching subjects with high societal impact such as precision farming or climate change. Furthermore, the selected tasks are sufficiently challenging for existing data science techniques, to the ill-posedness of involved simulation inversion. Knowledge gained from studying these tasks should be transferable to tasks in other domains as well, as long as they incorporate a somewhat similar way of generating simulation data. Finally, each of these tasks can be classified as a regression problem, has a measurable target parameter and ground truth data to evaluate with.

This benchmark set of tasks and data sources provides researchers with ready-to-use, preprocessed data from three sources: simulation models, in situ measurements and remote sensing observations. As a result, this potentially saves users of this benchmark from performing very time-consuming tasks including finding publicly available data, dealing with data quality issues, preprocessing the data and transforming the data into a format that can directly be used in a machine learning pipeline. The overall process used to obtain these ready-to-use datasets will be discussed further in Section 6.2.1. In Sections 6.1.1 to 6.1.4, we will discuss these tasks and data sources in more detail.

6.1.1 Leaf area index

Leaf area index (LAI) is an often monitored measurement that describes the density of leaves in a canopy. This applies to forest canopies, but also to other cover types such as croplands or grasslands. It is defined as the ratio of leaf area in comparison to ground area [64]. Leaf area index is often used as inputs to complex simulation models such as climate models. Therefore, accurate estimation of leaf area index has direct positive impact on other related scientific fields such as climate modelling. As leaf area index can be obtained in a non-destructive way using digital hemispherical photography, it is a well studied biophysical variable. Nevertheless, it is hard to obtain leaf area index

measurements on a large scale due to the manual work involved. Therefore, many studies have focused on estimating leaf area index from remote sensing images [70] and it still is an active research area.

Many in situ datasets focus on a particular geographical region or a particular cover type. For a benchmark set however, it would be interesting to have a broader range of cover types and geographical locations represented in the data. This way, any conclusions based on the benchmark data are transferable to a wider range of other datasets.

In situ data
The first data source we have chosen for this benchmark is the "Ground-Based Observations for Validation" (GBOV) project [4], which is part of the Copernicus Global Land Service. The aim of this project is to create a database of quality-controlled in situ observations over several different sites and spanning multiple years. With these observations, existing remote sensing products can be validated. Within this project, they offer two services: reference measurements (or in situ data) and land products. Here, the in situ data consist of ground truth data measured by physical sampling and field work. The land products contain estimates based on machine learning models trained on satellite data and reference measurements.

For our experiments, we use the reference measurements of leaf area index (RM7) [71]. This set of reference measurements encompasses twenty sites in the United States with varying cover types. These cover types include the following types as defined in the United States National Land Cover Database (NLCD) [38]: deciduous forest, cultivated crops, evergreen forest, mixed forest, shrublands, pasture, grassland and wetlands. The distribution of these cover types, as well as that of leaf area index values can be found in Figure 13.

Simulation data
Besides the in situ data from the GBOV database, we also use a source of simulation data for this task. For this purpose, we use the PROSAIL radiative transfer model [44]. The PROSAIL model is a widely used radiative transfer model in the field of plant cover monitoring [6]. It is typically used to retrieve biophysical parameters from remote sensing observations through the use of simulation model inversion. Examples of use cases include: estimating leaf area for different crops [19], monitoring grasslands [3] and estimating potato chlorophyll content [8]. It has been found to be suitable for parameter retrieval from satellite data and is computationally less demanding than other more advanced radiative transer models [6]. The PROSAIL model is composed of two submodels: PROSPECT and SAIL, where PROSPECT models light interactions at the leaf level [45] and SAIL at the canopy level [97].

Using PROSAIL, we generate a lookup table consisting of 100.000 unique simulations. The input parameters for these simulations were set using a Latin hypercube sampling approach [89] using the minimum and maximum values given in Table 7.

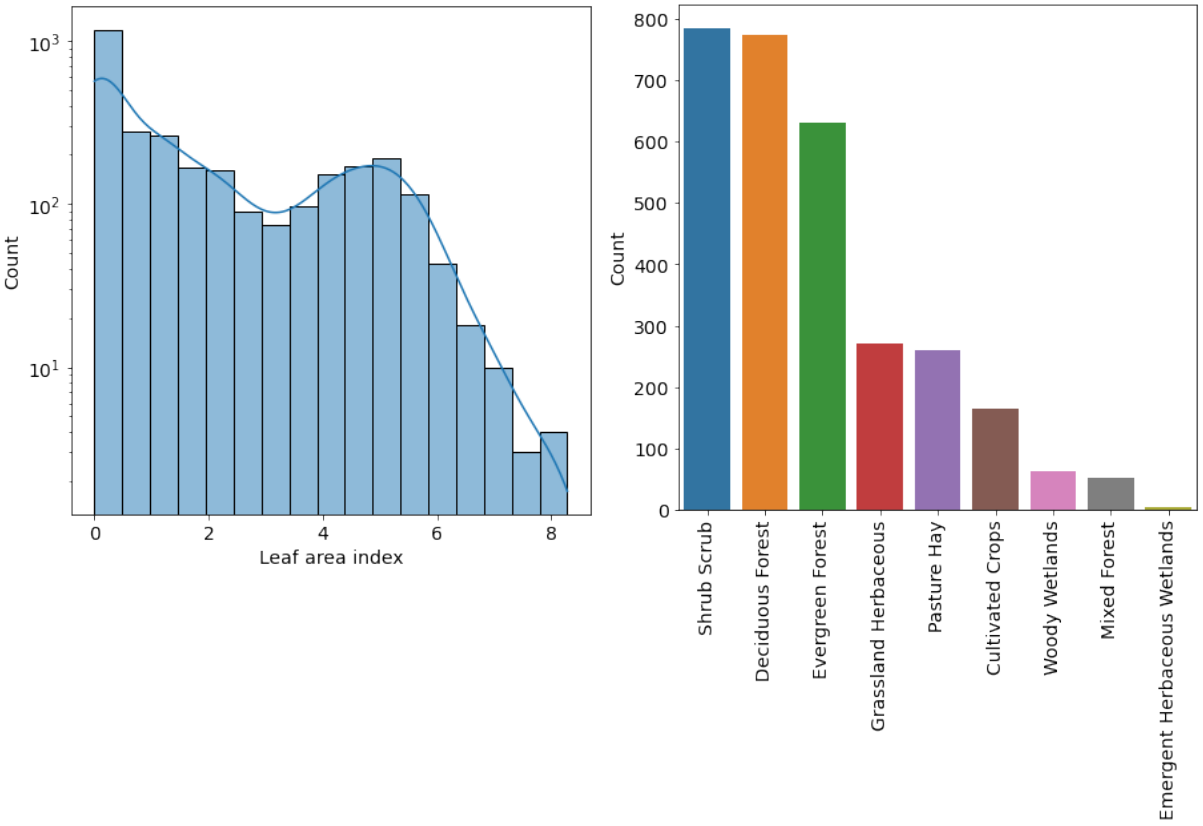


Figure 13: Left: Distribution of leaf area index. Right: Distribution of NLCD cover types.

Parameter	Symbol	Unit	Minimum	Maximum
Chlorophyll content	C_{ab}	μg cm^{-2}	20	80
Carotenoids content	C_{ar}	μg cm^{-2}	1	16
Brown pigment content	C_{bp}	Arbitrary units	0	2
Water equivalent thickness	C_w	cm	0	0.2
Dry matter content	C_m	g cm^{-2}	0	0.2
Leaf structure index	N	Arbitrary units	1.2	2.5
Leaf area index	LAI	m^2 m^{-2}	0	8
Average leaf inclination angle	ALIA	Degrees	5	85
Hot spot size	Hspot	Arbitrary units	0.05	0.1
Soil moisture	p_{soil}	Arbitrary units	0.4	0.5
Soil brightness	r_{soil}	Arbitrary units	0	1

Table 7: PROSAIL parameters used in leaf area index experiments.

Remote sensing data
Recent leaf area index studies often use the Sentinel-2 satellites [7] as remote sensing instruments. The Sentinel-2 remote sensing data is available starting from 23-06-2015. As most of the reference measurements were sampled after this date, we opted to use the Sentinel-2 level-1C top-of-atmosphere reflectance that is hosted on the Google Earth Engine platform [28].

Generally, Sentinel-2 data is processed into ready to use data using several steps: data selection, data downloading, cloud removal and atmospheric correction [74]. Where possible, we used Google Earth Engine. Using Google Earth Engine, it is possible to perform a majority of these processing steps server-side, reducing the amount of data transferred.

From the Google Earth Engine Sentinel-2 dataset [13], we filtered cloudy pixels based on the QA60 (cloud mask) band. This band contains a bitstring describing whether the pixel contains cirrus clouds, opaque clouds or is cloud-free. Only cloud-free pixels were considered.
Furthermore, we only considered remote sensing images that were taken within 15 days before or after the in situ sampling date. Any in situ samples that did not have a cloud-free overlapping pixel within this time limit were dropped from the data. After applying these filters, the dataset consists of 3004 data samples.

However, this remote sensing dataset contains top-of-atmosphere values whereas PROSAIL simulated bottom-of-atmosphere data. Unfortunately, bottom-of-atmosphere data for Sentinel-2 is only available starting from 03-28-2017. As a significant portion of the reference measurements are taken before this date, we opted to use the open source atmospheric correction algorithm 6S to convert Sentinel-2 top-of-atmosphere data to bottom-of-atmosphere data [53]. This conversion was done using the Python 6S interface "Py6S" [100].

6.1.2 Above-ground biomass

In light of the changing climate, many studies are performed on the storage of carbon in biomass. The world's forests form a large carbon sink and play a vital role in climate change [65]. As a result, it is a worthy endeavour to research ways to monitor the state of forests worldwide. A number of countries perform yearly national forest inventory studies, which can be used for monitoring biodiversity and habitat modelling [93].

While leaf area index only describes the canopy of some type of plant cover, above-ground biomass includes all parts of the plant that grow above the ground (no roots). This makes this measure more informative, but also harder to obtain. Where leaf area index can be computed from hemispherical digital photography, above-ground biomass is usually sampled in a destructive way. As a result, it is harder and more expensive to obtain. Remote sensing techniques may be useful to estimate parameters such as the amount of biomass in forests while avoiding or limiting destructive sampling [77].

In situ data
In many countries, yearly surveys are performed to monitor the status of their forests. One of these surveys is Sweden's National Forest Inventory [2]. In this survey, every 5th year measurements of volume and area are taken at certain permanent sample plots. In addition, there are some temporary sample plots which are visited yearly.

From this collected field data, a subset that is intended for the calibration or validation of remote sensing applications is publicly available online [42]. In this field data, dry weight for different fractions of the tree is given. As a measure of above-ground biomass, we take the sum of these parts. This results in a distribution of target values as shown in Figure 14. This Figure shows that while the dataset is somewhat skewed towards lower values, it covers a wide range of above-ground biomass measurements.

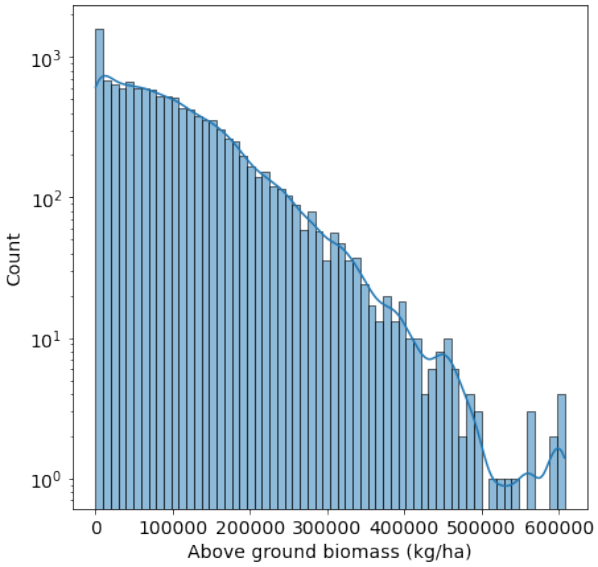


Figure 14: Distribution of above-ground biomass in dataset.

Simulation data
Similarly to the GBOV experiment, we use simulation data from the PROSAIL radiative transfer model. This model has been successfully used in biomass retrieval for different cover types, such as grasslands [32], croplands [99] and forests [104]. Again, we create a lookup table of 100.000 simulations using Latin hypercube sampling. As the parameterization of PROSAIL is highly dependent on the cover type, the parameterization slightly differs for the above-ground biomass experiment as shown in Table 8. The parameterization was based on previous works studying (boreal) forests [105, 106].

Parameter	Symbol	Unit	Minimum	Maximum
Chlorophyll content	C_{ab}	μg cm^{-2}	0	150
Carotenoids content	C_{ar}	μg cm^{-2}	0	37.5
Brown pigment content	C_{bp}	Arbitrary units	0.00001	8
Water equivalent thickness	C_w	cm	0	0.2
Dry matter content	C_m	g cm^{-2}	0	0.2
Leaf structure index	N	Arbitrary units	1	4.5
Leaf area index	LAI	m^2 m^{-2}	0	8
Average leaf inclination angle	ALIA	Degrees	10	89
Hot spot size	Hspot	Arbitrary units	0.05	0.1
Soil moisture	p_{soil}	Arbitrary units	0.5	1
Soil brightness	r_{soil}	Arbitrary units	0	1

Table 8: PROSAIL parameters used in above-ground biomass experiments.

Remote sensing data
For this experiment, we again use Sentinel 2 data. However, instead of using the level-1C Sentinel 2 dataset, we use the level-2A dataset which is already atmospherically corrected using the Sen2Cor algorithm [59]. The reason for choosing this source of satellite data is the availability of many in situ samples after 2017, which is the starting data for the Sentinel-2A dataset. Besides Sentinel-2 data, we also incorporate Sentinel-1 Synthetic Aperture Radar (SAR) data The SAR data was included as it less susceptible to signal saturation when dealing with thick canopies and is often used for biomass retrieval tasks [56, 76, 22]. This dataset comes pre-processed using the Sentinel-1 toolbox [96]. Both datasets are hosted on Google Earth Engine [28].

6.1.3 Ocean chlorophyll

The previous two subsections introduced datasets concerning the monitoring of land processes, specifically two problems (indirectly) related to the storage of carbon. Carbon can also be absorbed in the ocean by the photosynthesis process in phytoplankton [18]. Concentration of phytoplankton chlorophyll is commonly estimated using remote sensing, such as in the ESA Ocean Color climate Change Initiative (OC-CCI) [34]. These estimates are based on empirical approaches. However, these approaches are susceptible to significant errors of up to a factor of five [18]. Due to changing climate, these errors may increase as the composition of the ocean changes. As a result, the problem of estimating ocean chlorophyll concentration could be a very suitable candidate for introducing domain knowledge alongside data-driven methods.

In situ data
Within the scope of the ESA Ocean Color climate Change Initiative (OC-CCI) [34], multiple ocean-colour products were created. For the validation of these products, several

data sources are available. In 2016, Valente et al. created a compiled dataset from different ocean color validation data sources [94]. This compiled dataset contains both in situ measurements as well as remote sensing data. Chlorophyl-a concentrations were sampled using either high-performance liquid chromatography (HPLC) [86] or through fluorometric measurement [35]. As recommended by Valente et al., we use chlorophyll measurements from HPLC where available and supplement them with fluorometric measurements. This results in chlorophyll a concentrations ranging from 0 to 70 mg/m^3 as seen in Figure 15. Noticeably, a majority of the in situ measurements have a low chlorophyll concentration.

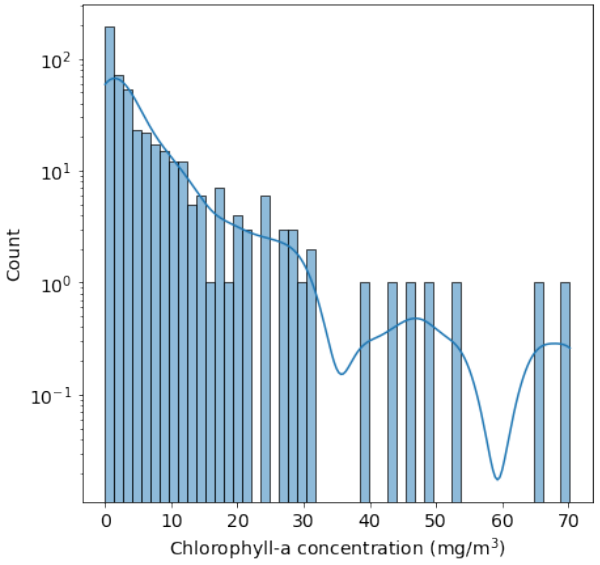


Figure 15: Distribution of chlorophyll a concentration in dataset.

Simulation data

To simulate the relationship between phytoplankton, chlorophyll a concentration and reflectance, we opted to use the HYDROPT open source framework [36] , based on the HydroLight radiative transfer model [62]. The HydroLight radiative transfer model is a commercial model that is frequently used in ocean chlorophyll simulation or retrieval tasks [67, 91, 102]. This framework is able to simulate concentrations of different phytoplankton size classes along with other colored disolved organic matter (CDOM).

As the creators of HYDROPT supply a lookup table of HYDROPT simulations [37] with their paper, we opted to use this lookup table instead of generating a new one. This lookup table contains different phytoplankton concentration levels for three different sized phytoplankton types: pico, nano and micro. Along with these measurements, the lookup table contains surface reflectance values. Even though phytoplankton concentration is equal to chlorophyll a concentration, they are highly correlated [17]. As our in situ data does not take into account differences in phytoplankton sizes, we take the average concentration of the three size classes for each simulated sample.

Remote sensing data
The compiled in situ dataset from Valente et al. [94] already includes reflectance values for different satellites, which are already linked to the in situ observations. These values are created by aggregating known measurements to other satellite bands within 6nm. We opted to use the satellite data with the highest coverage in terms of number of in situ samples, which turned out to be the ESA's Medium Resolution Imaging Spectrometer (MERIS) data. Here, we only used the first 7 bands of MERIS as the use of additional bands led to a very small number of available in situ samples.

6.1.4 Crop yield

Machine learning has become a valuable tool for crop yield prediction, supporting decisions on when to grow which crop type. Often used predictive features include soil type, precipitation, temperature and (remote sensing) images [95]. Typically, these features are taken over a time span instead of a single moment. Thus, in contrast to the other datasets introduced so far the problem of crop yield prediction does not take a static snapshot of physical parameters or satellite reflectances as input. Instead, it takes these features in a time window of a part of a season as input to predict the crop yield at the end of the season.

In situ data
Often yield measurements are proprietary data owned by farmers and not publicly available. The in situ data we use in this experiment was supplied by Michael Marszalek from ESA's Phi-lab. This dataset covers yield measurements for several crop types located in Bavaria (Germany) over the span of three years. For our experiment, we could only use a publicly available subset of the data, describing winter wheat yields in 2018. This subset is currently hosted on GitHub by ESA Phi-lab [68].

This dataset consists of geo-referenced yield measurements (ton/ha) from a combine harvester with a width of 13m. Based on the width of the combine harvester, we group the measurements in pixels of 10m by 10m. For each yield measurement (or pixel), a time-series of rain and evapotranspiration is supplied as additional features. These per-pixel time-series, along with remote sensing reflectance time-series, are aggregated into weekly features using linear interpolation where needed. This results in a 21-week long time-series for each yield measurement. As a final preprocessing step, all yield measurements that are more than 2 standard deviations removed for the mean are filtered out. This is typically done to remove incorrect measurements from combine harvesters that can occur on uneven terrain.

Simulation data
For this experiment, we again use the PROSAIL radiative transfer model as it is widely used to retrieve plant parameters in agricultural settings [15, 15]. Monitoring the growth status of crops is crucial for yield prediction. Even though PROSAIL cannot directly simulate crop yield, parameters used in the PROSAIL model can offer valuable insights

[90]. In order to capture relevant information for crop growth, we decided to invert PROSAIL for three different target parameters: leaf area index (LAI), water equivalent thickness (C_w) and dry matter content (C_m). For the creation of a lookup table, we used the PROSAIL parameters described in Table 9 These parameters were based on previous works studying PROSAIL for simulation of (winter) wheat [58, 6].

Parameter	Symbol	Unit	Minimum	Maximum
Chlorophyll content	C_{ab}	μg cm^{-2}	20	80
Carotenoids content	C_{ar}	μg cm^{-2}	1	16
Brown pigment content	C_{bp}	Arbitrary units	0	0.2
Water equivalent thickness	C_w	cm	0.0185	0.0185
Dry matter content	C_m	g cm^{-2}	0.002	0.2
Leaf structure index	N	Arbitrary units	1	1.8
Leaf area index	LAI	m^2 m^{-2}	0	8
Hot spot size	Hspot	Arbitrary units	0.05	0.1
Soil moisture	p_{soil}	Arbitrary units	0	1
Soil brightness	r_{soil}	Arbitrary units	0.5	0.6

Table 9: PROSAIL parameters used in crop yield experiments.

Remote sensing data

As the in situ measurements were taken in 2018, level-2A Sentinel 2 data is available. We chose to use this version of Sentinel-2 satellite data as it does not require further atmospheric correction, saving on computational time required. This dataset is included with the in situ data and available on ESA Phi-lab's GitHub [68]. Similarly to the rain and evapotranspiration data time-series, the satellite data time-series were resampled into a weekly frequency using linear interpolation where required.

6.2 Setup

Our experimental setup takes into account that for many Earth system science applications, it is non-trivial to gather in situ measurements. This is also reflected in the datasets used in our benchmark set. Each of these datasets only contains a few thousand in situ samples, which would be considered a very low amount for a typical machine learning setting. However, in practice these are quite large amounts of in situ data.

In order to have a more realistic view of model performance for real world scenarios, where in situ data availability is likely to be more limited, we perform experiments over an iteratively larger subset of training data. Based on the dataset characteristics given in Table 10, we chose to perform experiments for subsets of 50, 100, 250, 1000 and 2500 training samples.

Task	In situ samples	Simulation samples	Simulation model	Satellite data source	In situ target	Simulation target
Leaf area index (LAI)	3004	100,000	PROSAIL	Sentinel-2	LAI	LAI
Above-ground biomass (AGB)	11,973	100,000	PROSAIL	Sentinel-2, Sentinel-1 SAR	AGB	LAI
Ocean chlorophyll (Chla)	470	432	HYDROPT	MERIS	Chla	Chla
Crop yield	4887	100,000	PROSAIL	Sentinel-2	Yield	LAI, C_w, C_m

Table 10: Characteristics for each data source in the benchmark dataset.

6.2.1 Data processing

For each experiment, the data is processed in a similar way as visualized in Figure 16. As discussed in Section 6.1, we distinguish three data sources for each experiment: simulation data (from a simulation model), in situ data (real measurements) and satellite data (remote sensing).

The simulation dataset is built by creating a lookup table (LUT) from a set of combinations simulation parameters. These combinations are determined by Latin hypercube sampling, specifying the number of samples and a minimum and maximum value for each parameter. With Latin hypercube sampling, the sampled parameters are divided uniformly over the parameter space. This method of sampling was chosen as it has been used frequently in similar experiments [82, 61, 26].

The satellite data is first filtered based on quality flags such as cloud cover and timestamp, such that a satellite image is always taken within 30 days of an in situ measurement. On

this filtered data, atmospheric correction is applied. Afterwards, it is merged with the in situ data based on time and geographical coordinates. For each experiment run the merged in situ and satellite data is shuffled before splitting in training and holdout sets, where the shuffled order is different for each unique run.

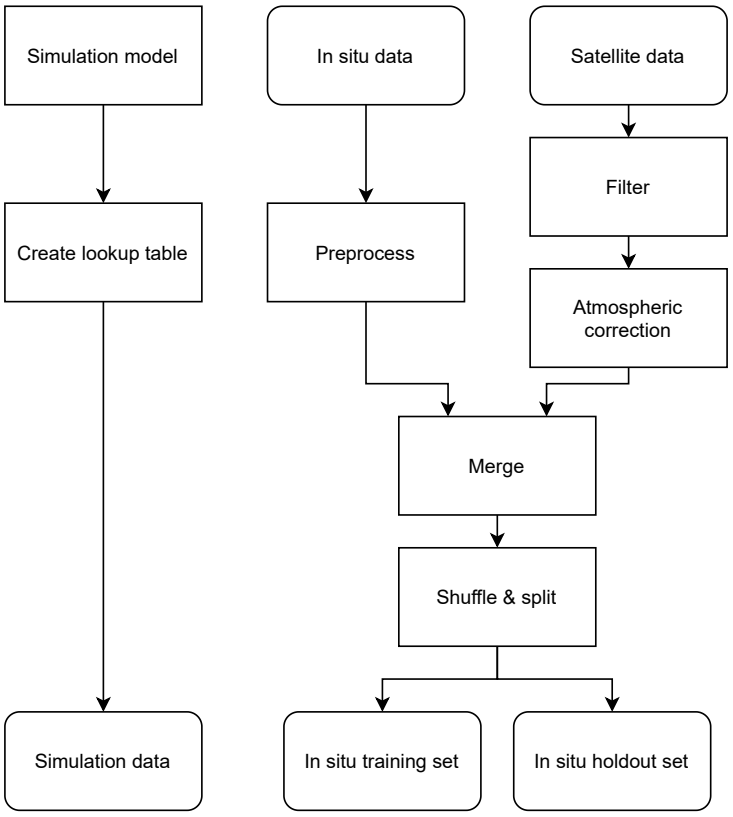


Figure 16: General overview of data processing pipeline.

6.2.2 Models

For each experiment, we use several data-driven models as baselines. These baselines are trained on either only in situ data, or only simulation data. These baselines consist of the models discussed in Section 4.2. Firstly, we have several simple baselines: Random forest, Gaussian process regression and Multilayer perceptron. These are optimized using Auto-sklearn, without ensembling. Secondly, we have a state-of-the-art baseline. For this baseline, we use Auto-sklearn without restricting which models can be chosen and with ensembling enabled. For these baseline models the Auto-sklearn settings can be found in Tables 11 and 12.

Parameter	Name	Value
Time per task	per_run_time_limit	18 minutes
Total time	time_left_for_this_task	3 hours
Ensemble size	ensemble_size	1
Initial meta learning configurations	initial_configurations_via_metalearning	1

Table 11: Auto-sklearn parameters used in all experiments for simple baseline models.

Parameter	Name	Value
Time per task	per_run_time_limit	18 minutes
Total time	time_left_for_this_task	3 hours
Ensemble size	ensemble_size	1
Initial meta learning configurations	initial_configurations_via_metalearning	1

Table 12: Auto-sklearn parameters used in all experiments for state-of-the-art baseline models.

6.3 Evaluation

As each experiment run only uses a subset of the in situ data as a training set, the rest of the data is available as a holdout set to use for evaluation. Within each experiment, hyperparameter tuning is performed using a subset of the training set. Each model uses 70% of the training data for model fitting and 30% of the training data for hyperparameter tuning (and model selection in Auto-sklearn). For the proposed method, the full set of available simulation data is always used. Again, 70% of the data is used for model fitting and 30% for tuning.

6.3.1 Metrics

To evaluate model performance, we use the R^2 metric as it is easy to interpret and compare between datasets The R^2 metric is also used as objective function in the Auto-sklearn framework for all our experiments. It represents the fraction of variance in the target value that is explained by the regression model. An R^2 score of 0 is achieved by predicting exactly the mean of the target values. An R^2 score of 1 is achieved by perfectly predicting the target values. The formula for R^2 is given in equation 3,

$$R^2 = \frac{SSE}{SST}, \tag{3}$$

where SSE denotes the sum of squares explained and SST the sum of squares total.

6.3.2 Bootstrapping

As the used modelling approaches are non-deterministic, we have to repeat our experiment several times in order to get a reliable set of results. As the experiments are computationally expensive to run, we employ a bootstrapping approach. For each dataset and for each model, we train and validate the model 15 times. Here, each repeated run uses a random seed equal to the run number. This way, it is ensured that different modelling approaches within the same run use the same subset of data (as the shuffling before splitting the data is random).

We derive a bootstrap distribution by repeatedly sampling 5 runs with replacement from the performed 15 runs. For each bootstrap sample, we can calculate statistics describing the sample, such as the mean rank per model. We repeat this process using 1000 repeats in total. From this distribution of statistics over all bootstrap samples, we can extract confidence intervals of our model performance for each experiment.

7 Results

In this chapter, we will compare the performance of several data-driven baseline with the performance of our proposed theory-guided framework. This comparison is made over a variety of training set sizes ranging from 50 to 2500 samples, to simulate various degrees of real world scarcity of in situ data. Where possible, we also experimented with 0 in situ training samples. In these cases, the models were trained on only simulation data and validated on the entire in situ dataset. This was only possible for the leaf area index and ocean chlorophyll retrieval tasks, as these are the only tasks where the simulation model estimates these parameters directly or with a very close proxy variable.

From our experiments, we want to answer the following questions:

1. Is there a relationship between the amount of available in situ data and the added value of incorporating simulation data into the modelling pipeline (e.g. does the proposed method work better at certain dataset sizes)?
2. Does any of the modelling techniques (from the baselines or proposed method) perform significantly better than the others?

7.1 Impact of available in situ data

To answer the first question, we will directly compare the performance of all models based on the R^2 metric, which describes which proportion of holdout data variance is explained by the model. Here a score of one denotes perfect predictions and a model that always predicts the average of the holdout set would get an R^2 score of zero. R^2 scores of below zero are also possible, these models create predictions that are worse than predicting the mean value. We will compare the R^2 score of each model over a variety of dataset sizes. In these experiments, all models were optimized by using the R^2 metric as objective function in the Auto-sklearn framework. The statistical significance of the results presented in Tables 13-16 was determined using the Python library Autorank [33].

	RF		GPR		MLP		Auto-sklearn		Hybrid		Hybrid-ensemble	
n_{train}	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
0	**0.44**	0.01	-1.13	0.00	-2.32	0.75	0.29	0.01	-	-	-	-
50	0.58	0.05	0.04	0.38	0.57	0.07	**0.62**	0.06	0.61	0.04	**0.63**	0.04
100	0.67	0.02	0.09	0.37	0.68	0.03	0.70	0.02	0.68	0.03	**0.72**	0.02
250	0.72	0.02	0.45	0.33	0.72	0.03	0.76	0.05	0.76	0.02	**0.78**	0.01
1000	0.83	0.01	0.35	0.35	0.85	0.01	**0.87**	0.00	0.84	0.01	**0.87**	0.00
2500	0.87	0.01	0.82	0.11	0.89	0.00	**0.90**	0.01	0.88	0.01	**0.90**	0.01

Table 13: Mean R^2 scores and standard deviation (std) calculated on a leaf area index in situ holdout set for different models and in situ training set sizes (n_{train}). A training set size of zero denotes training on only simulation data. Best performing model per training set size is marked in bold.

In Table 13 we see the results for our experiments on the leaf area index dataset. The proposed method, denoted as Hybrid and Hybrid-ensemble, is the only model that incorporates both simulation and in situ data. As can be expected, all models seem to improve when given more training data. Noticeably, the proposed method performs slightly better at very small in situ training set sizes, but is outperformed by the purely data-driven baselines at larger in situ training set sizes. Regarding the transferability of knowledge from simulation data to predicting in situ data, this is most likely the easiest task in the benchmark as the target value (leaf area index) can be directly simulated by the used simulation model (PROSAIL). In the other tasks, the target value cannot be directly simulated. Instead, we can simulate other biophysical parameters that are related to the target parameter. This way, we still retrieve some potentially informative features from the simulation model, especially if they are highly correlated with the target parameter.

Based on Table 14 the above-ground biomass task seems to be more difficult, particularly for smaller amounts of in situ training data. Only the Auto-sklearn baseline is able to consistently achieve an R^2 score of higher than zero for a training set size of 100. Above-ground biomass retrieval is generally regarded as a harder task than leaf area index retrieval, as it concerns the entire plant (or tree in this case) whereas leaf area only concerns the canopy. Satellite bands used in commonly used satellites such as Sentinel-2 can saturate for thick canopies. For this task, Sentinel-1 SAR data was also included, which is able to penetrate further into the canopy and does not saturate as quickly. However, the simulation models are not able to represent wavelengths used in SAR. If the Sentinel-1 features are more informative than the Sentinel-2 features, any theory-guided model will rely more heavily on only in situ data.

	RF		GPR		MLP		Auto-sklearn		Hybrid		Hybrid-ensemble	
n_{train}	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
50	-0.08	0.09	-0.06	0.04	-0.34	0.26	**0.01**	0.04	-0.23	0.25	-0.11	0.15
100	-0.01	0.05	-0.02	0.05	-0.08	0.07	**0.09**	0.02	-0.01	0.11	0.06	0.02
250	0.09	0.05	0.11	0.03	0.09	0.05	**0.16**	0.01	0.14	0.02	**0.16**	0.01
1000	0.19	0.01	0.20	0.02	0.20	0.01	**0.22**	0.00	0.20	0.01	0.21	0.00
2500	0.23	0.00	0.22	0.02	0.23	0.01	**0.23**	0.00	0.22	0.00	0.23	0.00

Table 14: Mean R^2 scores and standard deviation (std) calculated on an above-ground biomass in situ holdout set for different models and in situ training set sizes (n_{train}). Best performing model per training set size is marked in bold.

As the in situ dataset for the ocean chlorophyll estimation task was limited in size, we were only able to perform experiments with up to 250 in situ training samples. Furthermore, the size of the simulation data was limited as an existing lookup table was used instead of generating a new simulation dataset. In this task, the proposed theory-guided framework outperforms vanilla Auto-sklearn for the smallest training set size, but is outperformed by all baselines for larger training set sizes.

	RF		GPR		MLP		Auto-sklearn		Hybrid		Hybrid-ensemble	
n_{train}	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
0	-0.92	0.46	**-0.11**	0.03	-9.65	5.30	-8.30	3.07	-	-	-	-
50	0.22	0.05	0.27	0.06	0.24	0.22	**0.28**	0.06	0.22	0.07	0.27	0.07
100	0.32	0.08	0.39	0.07	**0.49**	0.07	0.46	0.05	0.36	0.06	0.40	0.08
250	0.57	0.04	0.59	0.04	0.57	0.08	**0.62**	0.04	0.44	0.13	0.51	0.07

Table 15: Mean R^2 scores and standard deviation (std) calculated on an ocean chlorophyll in situ holdout set for different models and in situ training set sizes (n_{train}). A training set size of zero denotes training on only simulation data. Best performing model per training set size is marked in bold.

n_{train}	RF mean	RF std	MLP mean	MLP std	Auto-sklearn mean	Auto-sklearn std	Hybrid mean	Hybrid std	Hybrid-ensemble mean	Hybrid-ensemble std
50	**0.13**	0.04	-0.06	0.08	0.05	0.08	0.08	0.09	0.11	0.10
100	0.23	0.06	0.17	0.07	**0.27**	0.04	-0.48	1.19	0.16	0.14
250	**0.38**	0.01	0.30	0.02	**0.38**	0.02	0.36	0.06	0.35	0.05
1000	0.50	0.00	0.45	0.01	**0.50**	0.00	0.49	0.00	**0.50**	0.01
2500	0.54	0.00	0.52	0.01	**0.55**	0.01	0.54	0.00	0.54	0.00

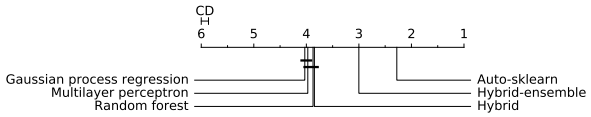
Table 16: Mean R^2 scores and standard deviation (std) calculated on a crop yield in situ holdout set for different models and in situ training set sizes (n_{train}). Best performing model per training set size is marked in bold.

The final task in our benchmark data is quite different from the first three tasks, as crop yield prediction uses a time window of features whereas the other tasks do not. In Table 16, we can see a clearly positive trend for model performance over increasing training set sizes. Noticeably, the proposed method with ensembling enabled outperforms all other models for the smallest dataset size. From a dataset size of 100 samples and upwards, the Auto-sklearn model appears to perform better than the other models. The Gaussian process regression baseline is not included in this experiment, as it consistently ran out of memory due to the large number of features used in this task.

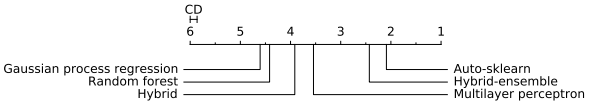
7.2 Model ranking

In order to answer the second question and determine whether any model is significantly better than the others, we can compare the models with each other in terms of ranking. Particularly, we are interested in whether the difference in model rankings is statistically significant. For this purpose, we created critical distance diagrams for different training set sizes in Figure 17 and for different datasets in Figure 18. These diagrams are based on the results of a Nemenyi post-hoc test as implemented by the autorank Python library [33]. From these Figures, we can conclude that the Auto-sklearn data-driven baseline is hard to beat. The proposed method with ensembling (Hybrid-ensemble) consistently ranks as second place, even at larger training set sizes. This could be explained by the fact that the data-driven part of the proposed method is very similar to the Auto-sklearn baseline. Thus, even in situations where the theory-driven part does not add much value, the data-driven part still allows for decent model performance. In specific circumstances the proposed theory-guided framework does outperform all baselines. This is for example true for the leaf area index dataset, as can be seen in Figure 18a. Another noticeable trend is that while Random Forest seems to perform better relative to other models for smaller training sets, Multilayer perceptron and Gaussian process regression perform better for larger training sets. Likely, Random forest is less sucsceptible to overfitting

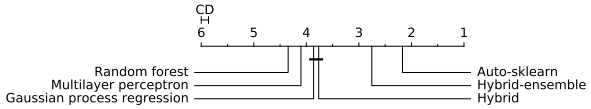
than Multilayer perceptron and Gaussian process regression. The risk of overfitting is higher for smaller training set sizes, and decreases as the training size increases.



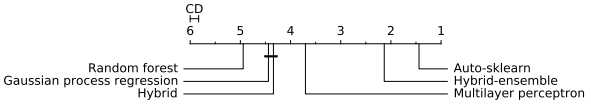
(a) 50 in situ samples



(b) 100 in situ samples

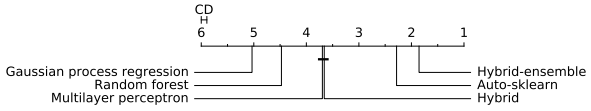


(c) 250 in situ samples

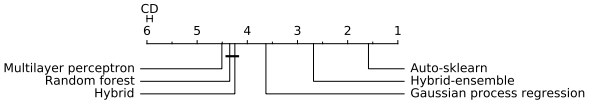


(d) 1000 in situ samples

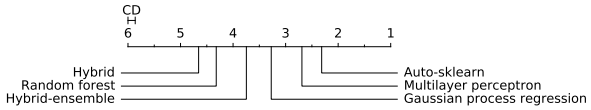
Figure 17: Critical distance diagrams for different training set sizes, values represent ranking and items connected by a vertical line are not significantly different.



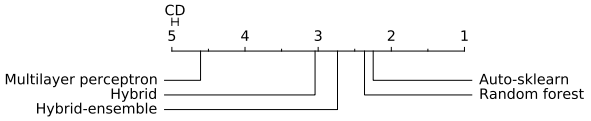
(a) Leaf area index



(b) Above-ground biomass



(c) Ocean chlorophyll



(d) Crop yield

Figure 18: Critical distance diagrams for different datasets, values represent ranking and items connected by a vertical line are not significantly different.

8 Conclusions & future work

In this thesis, we have explored the use of a theory-guided framework for several Earth system science tasks. We compared this framework to several data-driven baselines. For this comparison, we introduced a benchmark dataset consisting of four different Earth system science problems. This benchmark includes real world "in situ" data, as well as remote sensing and simulation data.

Our experimental setup was based on the scarcity of in situ data in real world Earth system science problems. Therefore, we have experimented with a range of differently sized subsets of our in situ datasets to reproduce scenarios where in situ data is difficult or expensive to gather. For each task in our benchmark data, our experimental results showed a clear positive relationship between number of in situ data samples used for training and model performance.

The effect of adding simulation data into our proposed theory-guided framework seemed limited, especially for experiments with larger subsets of the in situ dataset. Only for the smallest subsets there was some merit in incorporating simulation data into the modelling pipeline. The impact of adding simulation data also varied per task. For the leaf area index task, the impact was most pronounced. A likely reason is that this is the only task where the simulation model could directly model the target parameter. For all other tasks, the simulation models could only model proxy data. For example, in the ocean chlorophyll task, the simulation model could create estimates of phytoplankton concentration but not directly for chlorophyll a concentration. Another interesting result is that the proposed method performed very well for the yield prediction task at low dataset sizes. Apparently, the model is able to learn from simulation data if the amount of in situ data is limited. However for larger dataset sizes in the yield prediction task, it seems that simulation data does not add much value and the purely data-driven models perform better.

Another problem with introducing simulation data is that in the current implementation, it increases the dimensionality of the data. Especially for datasets of limited size, increasing dimensionality can result in overfitting and decreased generalization performance for models. On the other hand, problems where in situ data is hardly available can potentially have the largest boost in performance from introducing simulation data, as the information that can be extracted from a small dataset is limited and could be supplemented with information extracted from simulation data. It seems this trade-off between increasing dimensionality and extracting useful extra information from simulation data differs per task. Possibly, this also depends on the quality of the simulation model parameterization. Simulation model parameterization was not included in our modelling pipeline. Instead, it was based on existing work on similar problem settings (e.g., simulation of similar plant cover type).

8.1 Future work

In our current experiments, the parameterization of simulation models was not done by tuning it with available in situ data. Instead, it used parameterization from published works on similar problems. The impact of correct parameterization of a simulation model can be substantial. Therefore, a logical avenue for further research is the inclusion of simulation model parameterization into the modelling pipeline. This could be represented as extra hyperparameters in a CASH problem search space, similar to our current proposed framework, and could be a good candidate for incorporating automated machine learning techniques. Potential design of such a system is shown in Figure 19, where the parameterization of the simulation model is incorporated within the first (theory-driven) step of our proposed method. This method addresses the trade-off between increasing dimensionality and extracting useful information by improving the quality of the data supplied by the simulation model through proper parameterization of the model.

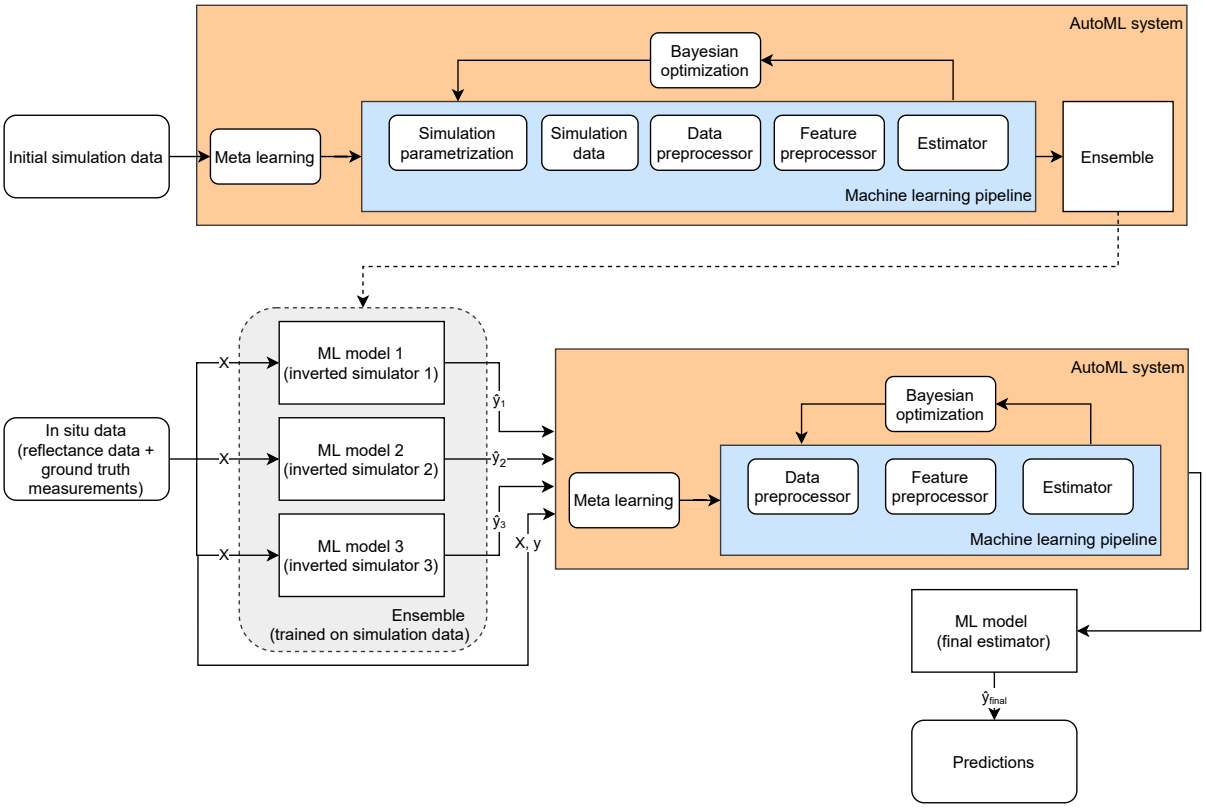


Figure 19: Parameterization of simulation model incorporated within the framework of our proposed method.

For each experiment, we only used a single simulation model even though different simulation models are available. In our proposed theory-guided framework, we use an ensemble of inverted simulation models. In order to capture a larger variety of information from

simulation models, it would be interesting to build this ensemble of inverted simulators from different types of simulation models. For example, for radiative transfer models there is often a difference in how the radiative transfer process is approximated in the model (e.g., a two stream or four stream equation).

Where the majority of tasks in our benchmark data were static problems where each feature is taken at a specific point in time, crop yield estimation usually takes as input a set of features that change over time. For example, weekly leaf area index measurements over the course of a season could be used as features. However, in this problem we still used a simulation model that simulates a static representation of parameters and not a time series. Simulation models that create time series for crop yield do exist, namely crop growth models. An opportunity for future research could be the parameterization of crop growth models using both static simulation data and (automated) machine learning and including this crop growth model along with in situ data into a modelling pipeline. Some work has already been done in combining PROSAIL with crop growth models [103, 39]. Parameterization of these model combinations using AutoML techniques could build further upon these works.

Lastly, our benchmark dataset only contains four distinct tasks where each task also only has a limited number of in situ data samples available. Four datasets is quite a limited amount for a benchmark set. Adding new data sources and new different tasks would improve the benchmark quality drastically. It would be even more interesting if new tasks could include forward problem settings as well seeing as all current tasks rely on simulation model inversion. An example of such a forward problem setting could be lake temperature estimation [49], where the used simulation model directly models the target parameter and does not need to be inverted first.

In conclusion, the results of experiments performed in this thesis using an automated framework to combine simulation- and machine learning models show that for certain cases, there is value in adding a theory-driven component to a modelling pipeline. However, there is a trade-off between increased dimensionality and information gained by the use of theory-guided methods. Future work on theory-guided methods should investigate their application to other Earth science tasks, or investigate ways to address this trade-off by either reducing dimensionality or increasing the amount of information gained from theory-driven components.

References

[1] R. Adeogun. "Calibration of Stochastic Radio Propagation Models Using Machine Learning". In: *IEEE Antennas and Wireless Propagation Letters* 18.12 (2019), pp. 2538–2542. DOI: 10.1109/LAWP.2019.2942819.

[2] Swedish University of Agricultural Sciences. *SLU National Forest Inventory*. 2021. URL: https://www.slu.se/en/Collaborative-Centres-and-Projects/the-swedish-national-forest-inventory/ (visited on 09/25/2021).

[3] Clement Atzberger et al. "Suitability and adaptation of PROSAIL radiative transfer model for hyperspectral grassland studies". In: *Remote sensing letters* 4.1 (2013), pp. 55–64.

[4] Gabriele Bai et al. "GBOV (Ground-Based Observation for Validation): A Copernicus Service for Validation of Vegetation Land Products". In: *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2019, pp. 4592–4594.

[5] Mariana Belgiu and Lucian Drăguţ. "Random forest in remote sensing: A review of applications and future directions". In: *ISPRS journal of photogrammetry and remote sensing* 114 (2016), pp. 24–31.

[6] Katja Berger et al. "Evaluation of the PROSAIL model capabilities for future hyperspectral model environments: A review study". In: *Remote Sensing* 10.1 (2018), p. 85.

[7] Michael Berger et al. "ESA's sentinel missions in support of Earth system science". In: *Remote Sensing of Environment* 120 (2012), pp. 84–90.

[8] Elizabeth J Botha et al. "Non-destructive estimation of potato leaf chlorophyll from canopy hyperspectral reflectance using the inverted PROSAIL model". In: *International Journal of Applied Earth Observation and Geoinformation* 9.4 (2007), pp. 360–374.

[9] Max Bramer. "Ensemble classification". In: *Principles of Data Mining*. Springer, 2013, pp. 209–220.

[10] Leo Breiman. "Bagging predictors". In: *Machine learning* 24.2 (1996), pp. 123–140.

[11] Manuel Campos-Taberner et al. "Multitemporal and multiresolution leaf area index retrieval for operational local rice crop monitoring". In: *Remote Sensing of Environment* 187 (2016), pp. 102–118.

[12] Gustau Camps-Valls et al. "Living in the Physics and Machine Learning Interplay for Earth Observation". In: *arXiv preprint arXiv:2010.09031* (2020).

[13] Earth Engine Data Catalog. *Sentinel-2 MSI: MultiSpectral Instrument, Level-1C*. 2021. URL: https://developers.google.com/earth-engine/datasets/catalog/COPERNICUS_S2 (visited on 11/05/2021).

[14] Subrahmanyan Chandrasekhar. *Radiative transfer*. Courier Corporation, 2013.

[15] Martin Danner et al. "Fitted PROSAIL parameterization of leaf inclinations, water content and brown pigment content for winter wheat and maize canopies". In: *Remote Sensing* 11.10 (2019), p. 1150.

[16] J. Deng et al. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[17] Blanka Desortová. “Relationship between chlorophyll-α concentration and phytoplankton biomass in several reservoirs in Czechoslovakia”. In: *Internationale Revue der gesamten Hydrobiologie und Hydrographie* 66.2 (1981), pp. 153–169.

[18] Heidi M Dierssen. “Perspectives on empirical approaches for ocean color remote sensing of chlorophyll in a changing climate”. In: *Proceedings of the National Academy of Sciences* 107.40 (2010), pp. 17073–17078.

[19] Si-Bo Duan et al. “Inversion of the PROSAIL model to estimate leaf area index of maize, potato, and sunflower fields from unmanned aerial vehicle hyperspectral data”. In: *International Journal of Applied Earth Observation and Geoinformation* 26 (2014), pp. 12–20.

[20] Imme Ebert-Uphoff et al. “A vision for the development of benchmarks to bridge geoscience and data science”. In: *17th International Workshop on Climate Informatics*. 2017.

[21] Katharina Eggensperger et al. “Towards an empirical foundation for assessing bayesian optimization of hyperparameters”. In: *NIPS workshop on Bayesian Optimization in Theory and Practice*. Vol. 10. 3. 2013.

[22] S Englhart, V Keuck, and F Siegert. “Aboveground biomass retrieval in tropical forests—The potential of combined X-and L-band SAR data use”. In: *Remote sensing of environment* 115.5 (2011), pp. 1260–1271.

[23] Matthias Feurer et al. “Efficient and robust automated machine learning”. In: *Advances in neural information processing systems*. 2015, pp. 2962–2970.

[24] Matthias Feurer et al. *Extending auto-sklearn 0.14.0*. 2021. URL: https://automl.github.io/auto-sklearn/master/extending.html (visited on 09/25/2021).

[25] Matt W Gardner and SR Dorling. “Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences”. In: *Atmospheric environment* 32.14-15 (1998), pp. 2627–2636.

[26] José Luis Gómez-Dans, Philip Edward Lewis, and Mathias Disney. “Efficient emulation of radiative transfer codes using Gaussian processes and application to land surface parameter inferences”. In: *Remote Sensing* 8.2 (2016), p. 119.

[27] Kasthurirangan Gopalakrishnan et al. “Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection”. In: *Construction and building materials* 157 (2017), pp. 322–330.

[28] Noel Gorelick et al. “Google Earth Engine: Planetary-scale geospatial analysis for everyone”. In: *Remote sensing of Environment* 202 (2017), pp. 18–27.

[29] Matthieu Guillaumin, Daniel Küttel, and Vittorio Ferrari. “ImageNet Auto-Annotation with Segmentation Propagation”. In: *International Journal of Computer Vision* 110 (2014), pp. 328–348.

[30] Mark Hall et al. “The WEKA data mining software: an update”. In: *ACM SIGKDD explorations newsletter* 11.1 (2009), pp. 10–18.

[31] Geoffroy Hautier et al. "Finding Nature's Missing Ternary Oxide Compounds Using Machine Learning and Density Functional Theory". In: *Chemistry of Materials* 22.12 (2010), pp. 3762–3767.

[32] Li He et al. "Retrieval of grassland aboveground biomass through inversion of the PROSAIL model with MODIS imagery". In: *Remote Sensing* 11.13 (2019), p. 1597.

[33] Steffen Herbold. "Autorank: A Python package for automated ranking of classifiers". In: *Journal of Open Source Software* 5.48 (2020), p. 2173. DOI: 10.21105/joss.02173. URL: https://doi.org/10.21105/joss.02173.

[34] Rainer Hollmann et al. "The ESA climate change initiative: Satellite data records for essential climate variables". In: *Bulletin of the American Meteorological Society* 94.10 (2013), pp. 1541–1552.

[35] Osmund Holm-Hansen et al. "Fluorometric determination of chlorophyll". In: *ICES Journal of Marine Science* 30.1 (1965), pp. 3–15.

[36] Tadzio Holtrop and Hendrik Jan Van Der Woerd. "HYDROPT: An Open-Source Framework for Fast Inverse Modelling of Multi- and Hyperspectral Observations from Oceans, Coastal and Inland Waters". In: *Remote Sensing* 13.15 (2021), p. 3006. ISSN: 2072-4292. DOI: 10.3390/rs13153006. URL: http://dx.doi.org/10.3390/rs13153006.

[37] Tadzio Holtrop and Hans van der Woerd. *HYDROPT: a Python framework for fast inverse modelling of multi- and hyperspectral ocean color data*. Version v0.1-alpha. May 2021. DOI: 10.5281/zenodo.4782707. URL: https://doi.org/10.5281/zenodo.4782707.

[38] Collin H Homer, Joyce A Fry, Christopher A Barnes, et al. "The national land cover database". In: *US Geological Survey Fact Sheet* 3020.4 (2012), pp. 1–4.

[39] Jianxi Huang et al. "Evaluation of regional estimates of winter wheat yield by assimilating three remotely sensed reflectance datasets into the coupled WOFOST–PROSAIL model". In: *European Journal of Agronomy* 102 (2019), pp. 1–13.

[40] F. Hutter, H. H. Hoos, and K. Leyton-Brown. "Sequential Model-Based Optimization for General Algorithm Configuration". In: *Proc. of LION-5*. 2011, 507–523.

[41] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.

[42] Swedish National Forest Inventory. *Sample plot data 2007-2020*. 2021. URL: https://www.slu.se/en/Collaborative-Centres-and-Projects/the-swedish-national-forest-inventory/listor/sample-plot-data/ (visited on 09/25/2021).

[43] Christopher Irrgang et al. "Will Artificial Intelligence supersede Earth System and Climate Models?" In: *arXiv preprint arXiv:2101.09126* (2021).

[44] S Jacquemoud et al. "Inversion de modèles de transfert radiatif pour estimer les caractéristiques d'un couvert végétal à partir de données de télédétection dans le domaine optique". In: *Comptes rendus du Coll. Int. La télédétection optique et radar et la géomatique pour la gestion des problèmes environnementaux* (1999), pp. 10–12.

[45] Stéphane Jacquemoud and Frédéric Baret. "PROSPECT: A model of leaf optical properties spectra". In: *Remote sensing of environment* 34.2 (1990), pp. 75–91.

[46] Xiaowei Jia et al. "Physics-guided machine learning for scientific discovery: An application in simulating lake temperature profiles". In: *arXiv preprint arXiv:2001.11086* (2020).

[47] Amogh Joshi. *Climate Reanalyzer — University of Maine — Today's Weather Maps*. 2021. URL: `https://climatereanalyzer.org/wx/DailySummary/` (visited on 10/31/2021).

[48] Amogh Joshi. *Next-Frame Video Prediction with Convolutional LSTMs*. 2021. URL: `https://keras.io/examples/vision/conv_lstm/` (visited on 10/31/2021).

[49] Anuj Karpatne et al. "Physics-guided neural networks (pgnn): An application in lake temperature modeling". In: *arXiv preprint arXiv:1710.11431* (2017).

[50] Anuj Karpatne et al. "Theory-guided data science: A new paradigm for scientific discovery from data". In: *IEEE Transactions on knowledge and data engineering* 29.10 (2017), pp. 2318–2331.

[51] Karthik Kashinath et al. "ClimateNet: an expert-labeled open dataset and deep learning architecture for enabling high-precision analyses of extreme weather". In: *Geoscientific Model Development* 14.1 (2021), pp. 107–124.

[52] Hong Seok Kim, Muammer Koc, and Jun Ni. "A hybrid multi-fidelity approach to the optimal design of warm forming processes using a knowledge-based artificial neural network". In: *International Journal of Machine Tools and Manufacture* 47.2 (2007), pp. 211–222. ISSN: 0890-6955.

[53] Svetlana Y Kotchenova et al. "Validation of a vector version of the 6S radiative transfer code for atmospheric correction of satellite data. Part I: Path radiance". In: *Applied optics* 45.26 (2006), pp. 6762–6774.

[54] M Kotila and A Mangal. *Shapes - Autonomio Deep Learning Workbench User Guide*. 2021. URL: `https://mikkokotila.github.io/slate/#shapes` (visited on 10/04/2021).

[55] Anders Krogh and Peter Sollich. "Statistical mechanics of ensemble learning". In: *Physical Review E* 55.1 (1997), p. 811.

[56] Thuy Le Toan et al. "Relating forest biomass to SAR data". In: *IEEE Transactions on Geoscience and Remote Sensing* 30.2 (1992), pp. 403–411.

[57] Songyang Li et al. "Combining color indices and textures of UAV-based digital imagery for rice LAI estimation". In: *Remote Sensing* 11.15 (2019), p. 1763.

[58] Zhenhai Li et al. "Remote sensing of leaf and canopy nitrogen status in winter wheat (Triticum aestivum L.) based on N-PROSAIL model". In: *Remote Sensing* 10.9 (2018), p. 1463.

[59] Jérôme Louis et al. "Sentinel-2 Sen2Cor: L2A processor for users". In: *Proceedings Living Planet Symposium 2016*. Spacebooks Online. 2016, pp. 1–8.

[60] Yanfei Lu et al. "Physics-Embedded Machine Learning: Case Study with Electrochemical Micro-Machining". In: *Machines* 5.1 (2017). ISSN: 2075-1702. URL: `https://www.mdpi.com/2075-1702/5/1/4`.

[61] Thomas Miraglio et al. “Joint Use of PROSAIL and DART for Fast LUT Building: Application to Gap Fraction and Leaf Biochemistry Estimations over Sparse Oak Stands”. In: *Remote Sensing* 12.18 (2020), p. 2925.

[62] Curtis D Mobley and Lydia K Sundman. “HYDROLIGHT 5 ECOLIGHT 5”. In: *Sequoia Scientific Inc* (2008).

[63] Mohammad Amin Morid, Alireza Borjali, and Guilherme Del Fiol. “A scoping review of transfer learning research on medical image analysis using ImageNet”. In: *Computers in biology and medicine* 128 (2021), p. 104115.

[64] Anthony Nguy-Robertson et al. “Green leaf area index estimation in maize and soybean: Combining vegetation indices to achieve maximal sensitivity”. In: *Agronomy Journal* 104.5 (2012), pp. 1336–1347.

[65] Yude Pan et al. “A large and persistent carbon sink in the world’s forests”. In: *Science* 333.6045 (2011), pp. 988–993.

[66] Bohdan Pavlyshenko. “Using stacking approaches for machine learning models”. In: *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*. IEEE. 2018, pp. 255–258.

[67] Marcela Pereira-Sandoval et al. “Calibration and Validation of Algorithms for the Estimation of Chlorophyll-A in Inland waters with Sentinel-2”. In: *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2018, pp. 9276–9279.

[68] ESA Phi-lab. *WorldCrops*. 2021. URL: `https://github.com/ESA-PhiLab/WorldCrops/tree/main/data/cropdata/Bavaria` (visited on 10/03/2021).

[69] Albert Porcar-Castell et al. “Linking chlorophyll a fluorescence to photosynthesis for remote sensing applications: mechanisms and challenges”. In: *Journal of experimental botany* 65.15 (2014), pp. 4065–4095.

[70] John C Price. “Estimating leaf area index from satellite data”. In: *IEEE Transactions on Geoscience and Remote Sensing* 31.3 (1993), pp. 727–734.

[71] Copernicus Global Land Products. *Ground-Based Observations for Validation (GBOV) of Copernicus Global Land Products - Reference Measurements*. 2021. URL: `https://land.copernicus.eu/global/gbov/products/` (visited on 09/09/2021).

[72] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational Physics* 378 (2019), pp. 686–707.

[73] Hassan Ramchoun et al. “Multilayer Perceptron: Architecture Optimization and Training.” In: *Int. J. Interact. Multim. Artif. Intell.* 4.1 (2016), pp. 26–30.

[74] Luigi Ranghetti et al. ““sen2r”: An R toolbox for automatically downloading and preprocessing Sentinel-2 satellite data”. In: *Computers & Geosciences* 139 (2020), p. 104473.

[75] Stephan Rasp et al. “WeatherBench: a benchmark data set for data-driven weather forecasting”. In: *Journal of Advances in Modeling Earth Systems* 12.11 (2020), e2020MS002203.

[76] Yrjo Rauste. "Multi-temporal JERS SAR data in boreal forest biomass mapping". In: *Remote sensing of environment* 97.2 (2005), pp. 263–275.

[77] Heather Reese et al. "Countrywide estimates of forest variables using satellite data and field data from the national forest inventory". In: *AMBIO: A Journal of the Human Environment* 32.8 (2003), pp. 542–548.

[78] Markus Reichstein et al. "Deep learning and process understanding for data-driven Earth system science". In: *Nature* 566.7743 (2019), pp. 195–204.

[79] Hongyu Ren et al. "Adversarial Constraint Learning for Structured Prediction". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, July 2018, pp. 2637–2643. DOI: 10.24963/ijcai.2018/366. URL: https://doi.org/10.24963/ijcai.2018/366.

[80] Juan Pablo Rivera et al. "Multiple cost functions and regularization options for improved retrieval of leaf chlorophyll content and LAI through inversion of the PROSAIL model". In: *Remote Sensing* 5.7 (2013), pp. 3280–3304.

[81] Marc Rußwurm and Marco Korner. "Temporal vegetation modelling using long short-term memory networks for crop identification from medium-resolution multi-spectral satellite images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2017, pp. 11–19.

[82] Nuno César de Sá et al. "Exploring the Impact of Noise on Hybrid Inversion of PROSAIL RTM on Sentinel-2 Data". In: *Remote Sensing* 13.4 (2021), p. 648.

[83] Francisco Sahli Costabal et al. "Physics-informed neural networks for cardiac activation mapping". In: *Frontiers in Physics* 8 (2020), p. 42.

[84] Eric Schulz, Maarten Speekenbrink, and Andreas Krause. "A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions". In: *Journal of Mathematical Psychology* 85 (2018), pp. 1–16.

[85] Hoo-Chang Shin et al. "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning". In: *IEEE transactions on medical imaging* 35.5 (2016), pp. 1285–1298.

[86] Yuzo Shioi, Rumiko Fukae, and Tsutomu Sasa. "Chlorophyll analysis by high-performance liquid chromatography". In: *Biochimica et Biophysica Acta (BBA)-Bioenergetics* 722.1 (1983), pp. 72–79.

[87] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. "Practical bayesian optimization of machine learning algorithms". In: *Advances in neural information processing systems* 25 (2012).

[88] "Stacked generalization". In: *Neural Networks* 5.2 (1992), pp. 241–259. ISSN: 0893-6080.

[89] Michael Stein. "Large sample properties of simulations using Latin hypercube sampling". In: *Technometrics* 29.2 (1987), pp. 143–151.

[90] Bo Sun et al. "Retrieval of rapeseed leaf area index using the PROSAIL model with canopy coverage derived from UAV images as a correction parameter". In:

International Journal of Applied Earth Observation and Geoinformation 102 (2021), p. 102373.

[91] Carolina Tenjo et al. "A new algorithm for the retrieval of sun induced chlorophyll fluorescence of water bodies exploiting the detailed spectral shape of water-leaving radiance". In: *Remote Sensing* 13.2 (2021), p. 329.

[92] Chris Thornton et al. "Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms". In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2013, pp. 847–855.

[93] Erkki Tomppo et al. "Combining national forest inventory field plots and remote sensing data for forest databases". In: *Remote Sensing of Environment* 112.5 (2008), pp. 1982–1999.

[94] André Valente et al. "A compilation of global bio-optical in situ data". In: In supplement to: Valente, A et al. (2016): A compilation of global bio-optical in situ data for ocean-colour satellite applications. Earth System Science Data, 8(1), 235-252, https://doi.org/10.5194/essd-8-235-2016. PANGAEA, 2015. DOI: 10.1594/PANGAEA.862886. URL: https://doi.org/10.1594/PANGAEA.862886.

[95] Thomas van Klompenburg, Ayalew Kassahun, and Cagatay Catal. "Crop yield prediction using machine learning: A systematic literature review". In: *Computers and Electronics in Agriculture* 177 (2020), p. 105709. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2020.105709. URL: https://www.sciencedirect.com/science/article/pii/S0168169920302301.

[96] Luis Veci et al. "The sentinel-1 toolbox". In: *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE. 2014, pp. 1–3.

[97] Wouter Verhoef. "Light scattering by leaf layers with application to canopy reflectance modeling: The SAIL model". In: *Remote sensing of environment* 16.2 (1984), pp. 125–141.

[98] Jochem Verrelst et al. "Retrieval of vegetation biophysical parameters using Gaussian process techniques". In: *IEEE Transactions on Geoscience and Remote Sensing* 50.5 (2011), pp. 1832–1843.

[99] Liang Wan et al. "Unmanned aerial vehicle-based field phenotyping of crop biomass using growth traits retrieved from PROSAIL model". In: *Computers and Electronics in Agriculture* 187 (2021), p. 106304.

[100] Robin Timothy Wilson. "Py6S: A Python interface to the 6S radiative transfer model." In: *Comput. Geosci.* 51.2 (2013), p. 166.

[101] Changjiang Xiao et al. "Short and mid-term sea surface temperature prediction using time-series satellite data and LSTM-AdaBoost combination approach". In: *Remote Sensing of Environment* 233 (2019), p. 111358.

[102] Peng Xiu, Yuguang Liu, and Junwu Tang. "Variations of ocean colour parameters with nonuniform vertical profiles of chlorophyll concentration". In: *International journal of remote sensing* 29.3 (2008), pp. 831–849.

[103] L Zhang et al. "Estimating wheat yield by integrating the WheatGrow and PROSAIL models". In: *Field Crops Research* 192 (2016), pp. 55–66.

[104] Linjing Zhang, Zhenfeng Shao, and Chunyuan Diao. "Synergistic retrieval model of forest biomass using the integration of optical and microwave remote sensing". In: *Journal of Applied Remote Sensing* 9.1 (2015), p. 096069.

[105] Qingyuan Zhang et al. "Can a satellite-derived estimate of the fraction of PAR absorbed by chlorophyll (FAPARchl) improve predictions of light-use efficiency and ecosystem photosynthesis for a boreal aspen forest?" In: *Remote Sensing of Environment* 113.4 (2009), pp. 880–888. ISSN: 0034-4257. DOI: https://doi.org/10.1016/j.rse.2009.01.002. URL: https://www.sciencedirect.com/science/article/pii/S0034425709000133.

[106] Qingyuan Zhang et al. "Estimating light absorption by chlorophyll, leaf and canopy in a deciduous broadleaf forest using MODIS data and a radiative transfer model". In: *Remote Sensing of Environment* 99.3 (2005), pp. 357–371. ISSN: 0034-4257. DOI: https://doi.org/10.1016/j.rse.2005.09.009. URL: https://www.sciencedirect.com/science/article/pii/S0034425705003044.

[107] Lucas Zimmer, Marius Lindauer, and Frank Hutter. "Auto-pytorch tabular: Multi-fidelity metalearning for efficient and robust autodl". In: *arXiv preprint arXiv:2006.13799* (2020).

[108] Raúl Zurita-Milla, VCE Laurent, and JAE van Gijsel. "Visualizing the ill-posedness of the inversion of a canopy radiative transfer model: A case study for Sentinel-2". In: *International journal of applied earth observation and geoinformation* 43 (2015), pp. 7–18.