Opleiding Informatica

Evaluating AutoML methods on hybrid inversion of PROSAIL RTM on Landsat-7 data for AGB estimation

Lieuwe Rooijakkers (s2012820)

Supervisors:
Nuno De Mesquita César de Sá
Mitra Baratchi

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

2022-01-03

Abstract

Monitoring biophysical parameters of vegetation is important to understand spatial processes of ecosystems. Field monitoring of these parameters is logistically difficult and therefore Remote Sensing (RS) provides an appealing alternative to monitor them. In addition, RS provides near-daily and virtually global acquisitions. Nevertheless, scarcity of field data can hinder the training of advanced models using RS. This lack of field data can potentially be addressed by using physically based models known as Radiative Transfer Models (RTMs) that simulate the interaction of light with vegetation. These models can be used to generate surrogate data for the training of machine learning models. This combination of physically based models with nonparametric regression methods models is commonly called *hybrid regression*. However, these hybrid regression approaches mostly use traditional machine learning methods, where the researcher is expected to possess adequate knowledge of these methods. Even for machine learning experts obtaining a high-performant model can be a challenge.

AutoML methods promise to solve these hurdles by automating the machine learning process, and thus taking the human out of the process. We investigated the performance of two AutoML methods (AutoKeras and Auto-sklearn) and compared them against two traditional models (Gaussian process regression and Random Forest regression) by training and evaluating them on pure artificial inversion. Our artificial dataset was generated by sampling canopy and leaf properties using Latin Hypercube Sampling (LHS), generating associated spectral responses using the PROSAIL RTM and applying noise. Finally, the performance of the models was evaluated by predicting above ground biomass (AGB) values from Landsat 7 (ETM+) imagery and comparing these estimations against field measurements. We found that on artificial data, Auto-sklearn had the best performance (MAE of 0.0034, $\sigma = 0.0003$). Here the average MAE for the AutoML methods (0.003925) was lower than the average MAE for the traditional machine learning methods (0.00465). On the field dataset, the Gaussian process regressor had the best performance (MAE of 0.0514, $\sigma = 0.0002$). Here the average MAE for the AutoML methods (0.055175) was higher than the average MAE for the traditional methods (0.05245).

Contents

1 Introduction **3**
- 1.1 Related work . . . 5
- 1.2 Research objectives . . . 6
- 1.3 Research questions . . . 6
- 1.4 Research hypothesis . . . 6

2 Algorithms **7**
- 2.1 PROSAIL RTM . . . 7
  - 2.1.1 RTM inversion . . . 8
- 2.2 Machine learning algorithms . . . 9
  - 2.2.1 Traditional machine learning algorithms for PROSAIL RTM inversion . . . 9
  - 2.2.2 Automated Machine Learning . . . 10

3 Methodology **14**
- 3.1 Artificial data generation . . . 15
  - 3.1.1 PROSAIL parameters . . . 16
- 3.2 Field data . . . 18
  - 3.2.1 Study area . . . 18
  - 3.2.2 Data preparation . . . 18
- 3.3 Method-specific parameters . . . 20
  - 3.3.1 Traditional approach to RTM inversion . . . 20
  - 3.3.2 Automated Machine Learning . . . 21

4 Results **22**
- 4.1 Artificial data . . . 23
- 4.2 Field data . . . 24

5 Discussion **25**
- 5.1 Research questions & hypotheses . . . 28
- 5.2 User experience . . . 28

6 Conclusions and Further Research **30**

A Auxiliary results **37**
- A.1 Artificial data . . . 38
  - A.1.1 Scatterplots . . . 39
  - A.1.2 Relationship between training time and performance . . . 43
- A.2 Field data . . . 46
  - A.2.1 Scatterplots . . . 46
  - A.2.2 Relationship between training time and performance . . . 48
- A.3 Method-specific . . . 49
  - A.3.1 Random forests . . . 49
  - A.3.2 Gaussian process . . . 50
  - A.3.3 AutoKeras . . . 51

B AdaNet **54**

C Reproducibility **54**

Acknowledgements

Thanks to my supervisors, Nuno César de Sá and Mitra Baratchi for their guidance, support and conversations. Thanks to Tom Smeding for making me realise there was an error in the calculation of the RMSE metric.

Plots in this thesis were generated using the Matplotlib [32] and seaborn [74] libraries.

Variables

In the following table we show a list of the variables used in this thesis, their unit and a small description.

Table 1: Variables, their meaning and their unit that are used in this thesis.

Variable	Description	Unit
LAI	Leaf area index	
C_m	Dry matter content	g/cm^2
AGB	Above ground biomass (equal to LAI $\cdot$ C_m) [27]	g/cm^2

1 Introduction

In the field of ecology, RS is used for purposes ranging from determining the health of lakes, farmland, and soil of the land around windparks, to determining leaf traits.

Remote sensing (RS), the gathering of spatial information of the Earth's surface from a distance [11], brings opportunities to scientists in many fields, like meteorology [49], archeology [76], and ecology. In the field of ecology, RS is used for purposes ranging from etermining the health of lakes [13], farmland [79], soil of the land around windparks [60] to determining leaf traits [57, 37].

One of the use cases for remote sensing methods in ecology is to retrieve biophysical variables of vegetation [70]. A biophysical variable is defined as "any vegetation property that can be quantified, i.e. any pigments, chemical constituents, structural variables, but also variables related to plant photosynthesis, productivity or diseases" [70].

In our research we focus on the Above Ground Biomass (AGB). In specific contexts, monitoring the amount of AGB has been shown to be useful, for example to determine the dynamics of an ecosystem [54, 21], to track the vegetation to find diseases [59], and to track animal populations [72]. In the case of our research the primary interest is to study the dynamics of AGB in the Oostvaardersplassen (Section 3.2.1) to ensure that animals have enough food. In the last Decennia animals were becoming thin and were dying; the die-off percentage without supplementary feeding varied between 6% in 2008 and 34% in 2005 [65].

AGB estimation, in particular on homogeneous canopies (such as grasslands), has been successful with Remote Sensing methods using satellites [27]. RS methods have the advantages of providing the possibility to observe a large area, which is required to obtain good results in spatially heterogeneous environments in the context of canopy characteristics estimation [3]. They also have the advantage of being less expensive and time consuming, and furthermore they can be used to retroactively observe regions. Imagery from publicly funded remote sensing satellite missions, such as the Landsat and Sentinel missions, are readily available and free to use for any use.[1][2] However, the missions are limited in their resolution: the Landsat 7 and 8 missions both feature an image resolution of 30 meters[3], and the Sentinel-2 mission has bands with resolutions ranging from 10 to 60 meters[4]. The commercial WorldView-4 satellite boasts a sensor that has a resolution of 31cm for the panchromatic channelx and 124cm for the multispectral channel.[5] The imagery from the commercial satellites is generally not free to use.

Verrelst *et al.* [70] summarised the current approaches for biophysical variable retrieval methods from spectroscopy into four categories:

1. *Parametric regression methods*: Regression methods where an explicit relationship between spectral observations and specific biophysical variables is assumed. Therefore, parametrized expressions can be built. These methods require knowledge beforehand of the statistical relationship that exists between variables and spectral responses.

[1] Landsat 7 data access information: `https://developers.google.com/earth-engine/datasets/catalog/LANDSAT_LE07_C02_T1_L2#terms-of-use`

[2] Sentinel-2 data access information: `https://sentinel.esa.int/documents/247904/1848117/Sentinel-2_Data_Products_and_Access`

[3] `https://developers.google.com/earth-engine/datasets/catalog/LANDSAT_LE07_C02_T1_L2#bands`

[4] `https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/resolutions/spatial`

[5] `https://web.archive.org/web/20160424030451/https://dg-cms-uploads-production.s3.amazonaws.com/uploads/document/file/196/DG_WorldView4_DS_11-15_Web.pdf`

2. *Nonparametric regression methods*: Regression methods where no explicit relationship needs to be known beforehand between variables and spectral responses; the regression functions are defined directly by the information provided to the methods. The relationship between the variables and spectral responses is therefore made non-explicitly.

3. *Physically based model inversion methods*: Inversion methods that rely on techniques that model physical laws. They typically use Radiative Transfer Models (RTMs).

   In the field of RS a widely used RTM is PROSAIL [35]: an RTM model that combines the PROSPECT [34] and SAIL [66] models. PROSPECT uses the physical principles of light interaction with matter to simulate the optical properties of a single plant leaf. SAIL (*Scattering by Arbitrary Inclined Leaves*) works on an idealized form of a canopy, which in turn is used to simulate the canopy reflectance according to a single leaf reflectance [66]. In Section 2.1 a more detailed explanation of PROSAIL is given.

4. *Hybrid regression methods*: Hybrid regression methods combine nonparametric regression methods with physically based methods. They retain the flexibility and computational efficiency of nonparametric regression methods and the generic properties of physically based models.

Current research shows promising results using the hybrid regression method mentioned in the above list. A nonparametric model is used to train the inverse relationship between the spectral output of an RTM model (such as PROSAIL) to the biophysical variables to generate that output. Hybrid inversion technqiues can be used in situations were there is a limited amount of field data available, or were gathering field data would be expensive. However, inverting RTM models is still challenging due to the problem of ill-posedness: the same spectral profile can be obtained by different combinations of biophysical variables [23]. For the nonparametric and hybrid approaches we define two categories of machine learning methods:

Traditional machine learning methods being nonautomated nonparametric regression methods; examples are Support Vector Machines (SVMs), k-nearest neighbours, Random Forests, and Gaussian Processes. These methods require domain-specific knowledge in the field of data science, and even for experts they require substantial research and time to create well-performing models [28]. While these methods have produced promising results, not all ecologists are knowledgable enough in the field of data science and/or artificial intelligence. In our comparison we use Random forests [25] and Gaussian process [8] regressors for the Traditional methods, since they are commonly used in literature for biophysical variable estimation [15, 56, 22].

Automated machine learning (AutoML) methods are an increasingly promising solution to this obstacle. AutoML methods automate a larger portion of the AutoML pipeline (Fig. 3), thus allowing scientists without much statistical and/or machine learning knowledge to leverage machine learning in their research [28]. Examples of open-source AutoML methods are Auto-sklearn [18], AutoKeras [36], TPOT [45], and H2O [46]. Commercial offerings are also available, for example Google's Cloud AutoML[6] and Microsoft's Azure Automated machine learning[7]. In our research we use Auto-sklearn and AutoKeras for the Automated methods, since they are easy to use (Section 5.2), are popular, and automate a large portion of the AutoML pipeline (Fig. 3). We explore AutoML methods more in-depth in Section 2.2.2.

[6] `https://cloud.google.com/automl/`

[7] `https://azure.microsoft.com/en-us/services/machine-learning/automatedml/`

Because of the aforementioned issues with traditional machine learning methods and the improvements the AutoML techniques promise to bring, we will compare the mentioned frameworks to the problem of AGB estimation in a hybrid regression approach. We will evaluate the performance of the models on artificial data generated using the PROSAIL RTM. Then we will apply these models to estimate the AGB value from Landsat 7 imagery in the Oostvaardersplassen, comparing against field measurements done. Landsat 7 is a satellite that uses the Enhanced Thematic Mapper Plus (ETM+) image sensor. Landsat 7 imagery was chosen since the mission started in 1999, which means that we have images for the whole history of the field dataset at our disposal. The goal of our research is to conclude whether or not AutoML are useful to the problem of AGB estimation from spectral imaging using a hybrid approach, by evaluating whether or not they perform as well as, or better than the traditional methods.

1.1 Related work

Verrelst *et al.* [70] provide a review into retrieval methods; they reviewed all the methods we classified as traditional in our comparisons. Their review is of interest to our research since it lays the groundwork of definitions and the hybrid approach to which we evaluate the four machine learning methods.

Sá *et al.* [56] compared Artificial Neural Networks, Gaussian process regressors, Multi-task Neural Networks, and Random forests regressors for the hybrid inversion of the PROSAIL RTM. They evaluated the performance of the methods on artificial dataset, both with and without added noise. They found the Gaussian process regressor to be the best-performing when when no noise was added, and ANNs the best-performing when noise was added. We build on their approach for hybrid inversion, especially their choices for artificial noise generation. In our research we add evaluation on field data and also evaluate the performance of AutoML packages.

Estévez *et al.* [15] trained Gaussian process regressors in a hybrid approach using PROSAIL to predict LAI from Sentinel-2 data. They also trained using artificial noise and evaluated their performance on field data, which we also do. The only difference is that they predict LAI and evaluated the performance of the models on field data. We cannot do the latter since the field data that we have at our disposal does not include LAI measurements (Section 3.2). In Section 5 we make an approximate comparison between the findings of this thesis and that of Estévez *et al.*

Gao *et al.* [22] used a nonparametric approach using the Artificial Neural Network, Support Vector Machine, k-nearest neighbour, linear and Random forests regressors. Since Gao *et al.* used a nonparametric approach and we did an hybrid approach, it might be interesting to compare our findings, especially since both evaluate the Random forests regressor on the task of AGB estimation on LANDSAT data (albeit their dataset uses Landsat 5 imagery, while we use Landsat 7 data).

He *et al.* [27] used Lookup-table (LUT) inversion for a physically based inversion approach using the PROSAIL RTM to estimate the AGB variable from MODIS imagery. Their paper is of interest to us since we use their definition of AGB. It is also interesting to explore the difference in performance between the machine learning methods that we use against a physically based method (LUT inversion).

Verrelst *et al.* [69] used Lookup-table (LUT) inversion to estimate the LAI and C_{ab} variables in a physically based approach using the PROSAIL RTM applied on Sentinel-2 and -3 data. Just like Verrelst *et al.*, this thesis give us another baseline to compare the results we obtained using hybrid inversion to that of a physically based approach.

Truong *et al.* [64] compared Auto-sklearn, AutoKeras, auto_ml, H2Os AutoML, TPOT, Darwin, and Ludwig on various regression and classification datasets. They found Auto-sklearn, AutoKeras, and H2Os AutoML to be the best performing. this thesis is of interest since they compare both Auto-sklearn and AutoKeras for regression problems, which we also do in this thesis. Since this thesis also compares traditional methods on the same dataset, it has the opportunity to provide more insight in the performance of Auto-sklearn and AutoKeras.

Balaji and Allen [1] compared Auto-sklearn, TPOT, auto_ml and H2Os AutoML on various regression and classification datasets. They found that Auto-sklearn performs the best on regression problems compared to the other libraries tested. While less related to our research than the previous paper — they don't compare with Auto-sklearn — it still gives insight, especially when combining their results with the results of the previously mentioned paper.

1.2 Research objectives

Our research objectives are as follows:

1. Compare the performance of traditional machine learning methods used for hybrid inversion of PROSAIL RTM with AutoML methods for Landsat ETM+ PROSAIL RTM inversion.

2. Compare the performance of traditional methods with AutoML methods on real Landsat TM/ETM data.

1.3 Research questions

Main question: Do AutoML methods outperform traditional machine learning techniques when used as a hybrid regression model for the estimation of AGB by RTM inversion?

1. Do AutoML methods have a lower MAE than traditional machine learning methods when used as hybrid regression methods for RTM inversion of artificial spectral data?

2. Do AutoML methods have a lower MAE than traditional machine learning methods when used as hybrid regression methods for RTM inversion of field data?

1.4 Research hypothesis

Hypothesis 1 AutoML methods have a lower MAE than traditional machine learning methods when used as Hybrid regression methods for RTM inversion of artificial spectral data.

H_0 The MAE of AutoML methods is equal to or higher than traditional machine learning methods.

H_a The MAE of AutoML methods is lower than traditional machine learning methods.

Hypothesis 2 AutoML methods have a lower MAE than traditional machine learning methods when used as Hybrid regression methods for RTM inversion of Landsat data compared to field data.

H_0 The MAE of AutoML methods is equal to or higher than traditional machine learning methods.

H_a The MAE of AutoML methods is lower than traditional machine learning methods.

2 Algorithms

In this section a small introduction will be given for all the algorithms used and evaluated in this thesis to the problem of RTM inversion. We will introduce the PROSAIL, the RTM model used to generate an artificial dataset to train the models on. We will also introduce the methods we compare, all support both single-output regression and multi-output regression.

First we will introduce the traditional machine learning methods commonly used in literature for RTM inversion. The methods in this section that we will compare are: Random forests and Gaussian process.

Secondly we have the AutoML frameworks: Auto-sklearn and AutoKeras.

2.1 PROSAIL RTM

Radiative transfer models (RTMs) are physical models that describe the interaction between radiation and objects [70]. Although there are many different RTMs, our focus is in the interaction between sunlight radiation and vegetation which is parametrized by biophysical (e.g. LAI) and biochemical properties (e.g. Chlorophyll).

PROSAIL [35], first described by Baret *et al.* [2], is an RTM model that describes that interaction. It is the most common used RTM to retrieve biophysical variables, and therefore also the most used in hybrid regression methods [56]. PROSAIL works by combining PROSPECT [34] (a model for leaf spectra) with the SAIL [66] (*Scattering by Arbitrary Inclined Leaves*) model (a model for modelling the light scattering in canopies). PROSPECT uses the physical principles of light to matter interaction to simulate the optical properties of a single plant leaf. SAIL works on an idealized form of a canopy, it assumes the canopy to be [66]: 1. An infinite plane, 2. formed by small and flat leaves, 3. homogeneous. SAIL can then simulate the interaction between a light source and the canopy using the single leaf reflectance as generated by PROSPECT and calculate the final reflectance of light on the canopy. The output of the PROSAIL model is the simulated spectral profile of the canopy between the wavelengths of 400nm and 2500nm. This profile is described by the input parameters: leaf traits (e.g. C_{ab}), canopy traits (e.g. LAI), and sensor/sun positions (see Table 3). A schematic showing a general overview of the workings of PROSAIL is given in Fig. 1.

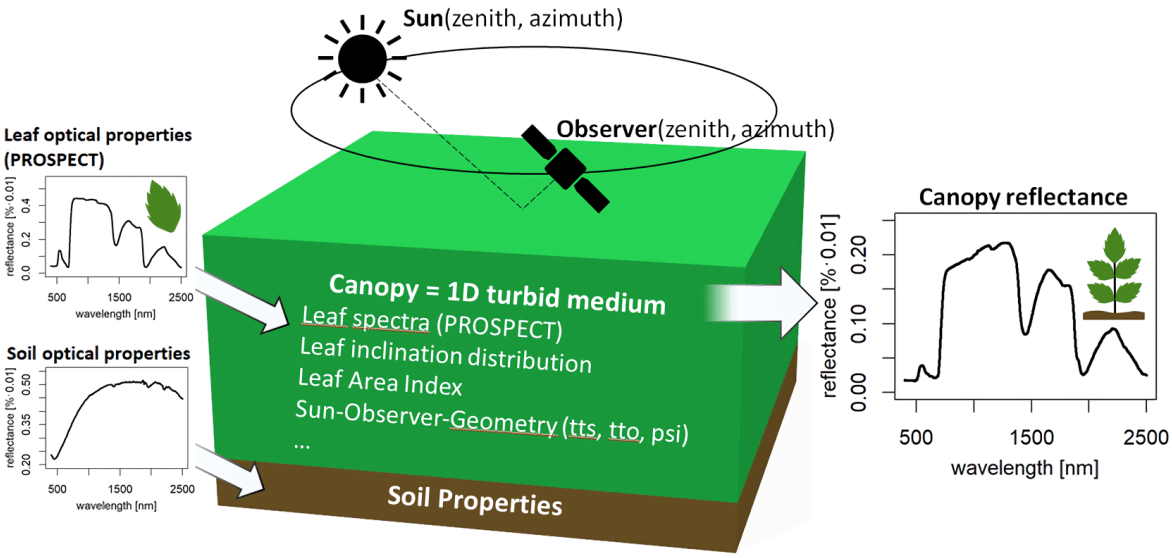


Figure 1: Schematic giving an overview of PROSAIL. By Kattenborn [38]

2.1.1 RTM inversion

We can use (non)parametric regression methods in junction with RTMs in two ways: *emulation* [68] and *inversion* [70]. The schematic Fig. 2 shows the relationship between emulation and inversion.

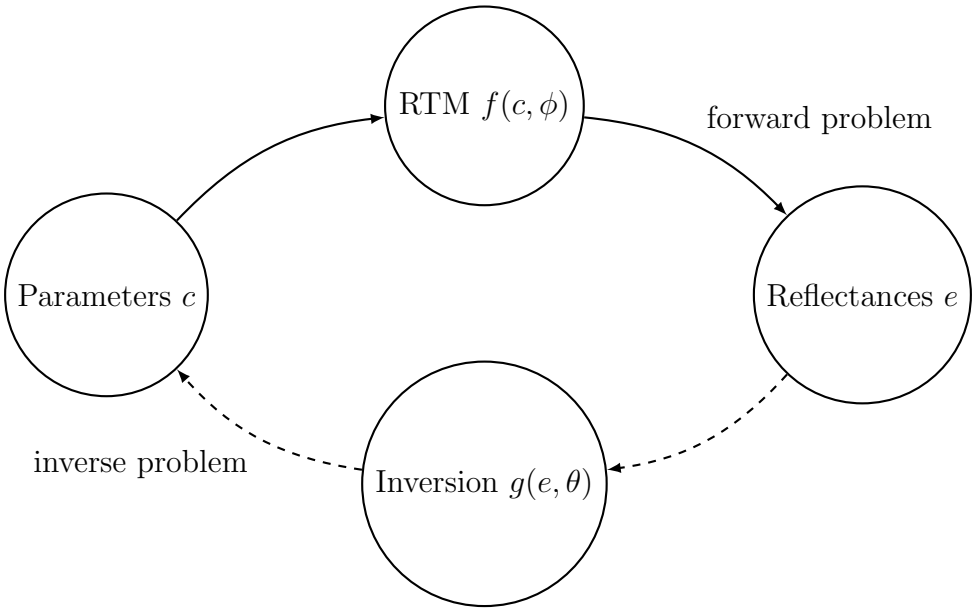


Figure 2: A schematic showing the relationship between the forward problem (*emulation*) and the inverse problem (*inversion*). In our research we are studying the inverse problem, learning a function g which estimates the parameters c for which the RTM f will again produce e. Adapted from Svendsen *et al.* [61]

Emulation consists of learning the relationship between the input variables to a RTM and the spectra it produces, thereby simulating the RTM model [68]. This can be interesting when

the RTM is expensive and time consuming to compute, since the emulated model would be less computationally expensive. Most RTMs are hard to compute because they are rigorous physically based models [23]. Learning an estimation of these RTMs can provide a way more performant model, which most of the times delivers results which are very close to the actual result [23].

Inversion consists of taking an RTM and learning the **inverse** relationship between (some of) the input variables and the output spectra (i.e. we take generated spectra and try to estimate the biophysical variables) [70]. Inversion can be used to efficiently estimate variables — such as AGB — from Remote Sensing data. The focus of this thesis is on estimating this inversion problem. RTM inversion is negatively impacted by ill-posedness, meaning that the same spectral profile can be obtained by different combinations of biophysical variables [23]. Complications occur because many biophysical variables have similar and drastic effects on the output spectra, which makes this inverse problem ill-posed [67]. Ill-posedness will increase in the case of a limited number of bands and with noisy data as is generally the case in RS data [82]. As increasing the amount of target parameters increases both the ill-posedness of the problem and exacerbates the curse of dimensionality we won't learn to predict all the input parameters, but only LAI and C_m (or in the case of single-output the product of the two).

2.2 Machine learning algorithms

This section provides a small introduction to the machine learning algorithms that were used. The section is divided into traditional and automated machine learning methods. The first is used to describe the more commonly used algorithms for RTM inversion, while the latter encompasses the two AutoML packages used in this research: AutoKeras and Auto-sklearn.

Orginally also the AdaNet [75] library was included, but due to the properties of the library we did not include it in the final comparison. We explain this in more detail in Appendix B.

2.2.1 Traditional machine learning algorithms for PROSAIL RTM inversion

Random forests

The general idea behind Random forests (RF, also known as random decision forests) was introduced by Tin Kam Ho in 1995 [31]. But it was Leo Breiman in 2001 [5] to extend and properly introduce Random forests.

The algorithm works by constructing a fixed amount (B) of uncorrelated decision trees. In Breiman this is done using the CART [10, 40] procedure. The trees are combined using Bootstrap aggregating (also known as bagging) which learns an ensemble of trees trained on different randomly sampled subsets.

Random forests also combine the random subspace method (as introduced by Tin Kam Ho). This method attempts to reduce the correlation between the learned trees in the forest. If there is a strong predictor in the dataset that explains the output variable, one can expect many of the B trees to also split on that predicting feature, thus leading to an unwanted correlation [30]. A solution is the random subspace method (also known as feature bagging), this method reduces the correlation not by randomly sampling datapoints from the dataset, but by sampling the features. This makes an individual tree not being able to predict accurately on the dataset, but since we create a forest we circumvent this problem and reduce the correlation between the trees, thus reducing variance.

Random forests can be used for regression or for classification. In the context of this thesis we use Random forests for regression, where the mean value of the all the predictions done by the individual estimators is used as the final prediction of the result. We use the Random forests implementation from the scikit-learn (sklearn) [53] library.

Gaussian process

Gaussian process regression is usually attributed to Matheron in 1962, where it was used in the geostatistics field [8]. But it also finds use in general statistics [51]. A Gaussian process model creates a posterior based on a prior distribution updated with observed datapoints from a dataset [71]. This probability distribution is a distribution over possible functions that fit the set of points the model is trained on. Since we are left with a probability function we can calculates the variances and the means, therefore deducing how confident we are about a prediction [19].

There are multiple methods to calculate this probability distribution. According to their documentation, sklearn uses algorithm 1.2 from [55], this uses Cholesky decomposition to aid in the performance and numerical stability. We use the Gaussian process implementation from the scikit-learn (sklearn) [53] library.

2.2.2 Automated Machine Learning

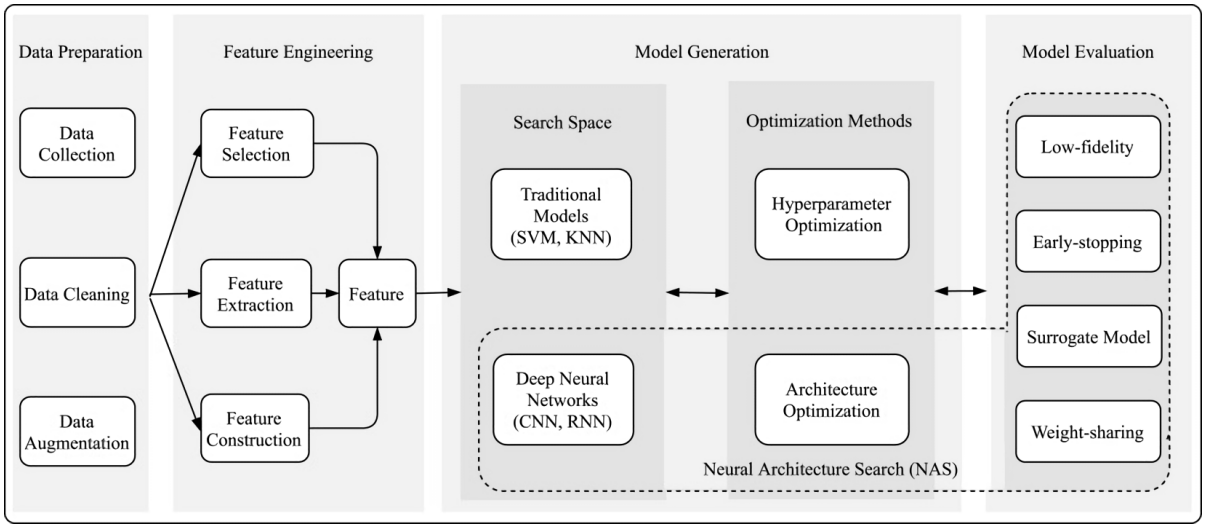


Figure 3: Overview of the AutoML pipeline. He, Zhao and Chu [28]

Automated Machine Learning (AutoML) encompasses the idea of automating the entire process of setting up, using, applying and maintaining the pipeline required for machine learning methods [28].

AutoML methods can be formalised as a Combined Algorithm Selection and Hyperparameter optimization (CASH) problem [63]. Given a set of algorithms $\mathcal{A} = A^{(1)}, \ldots, A^{(k)}$, with associated hyperparameter spaces $\Lambda^{(1)}, \ldots, \Lambda^{(k)}$, CASH is defined as computing:

$$A^*_{\lambda^*} \in \underset{A^{(j)} \in \mathcal{A}, \lambda \in \Lambda^{(j)}}{\operatorname{argmin}} \frac{1}{k} \sum_{i=1}^{k} \mathcal{L}(A^{(j)}_{\lambda}, \mathcal{D}^{(i)}_{\text{train}}, \mathcal{D}^{(i)}_{\text{valid}}) \tag{1}$$

Where $\mathcal{D}^{(i)}$ denotes the dataset i in k-fold cross-validation. Where $\mathcal{L}(A_\lambda^{(j)}, \mathcal{D}_{\text{train}}^{(i)}, \mathcal{D}_{\text{valid}}^{(i)})$ is the loss achieved by A when trained on $\mathcal{D}_{\text{train}}^{(i)}$ and evaluated on $\mathcal{D}_{\text{valid}}^{(i)}$. Thus finding the algorithm and hyperparameter set that results in the lowest mean loss value in k-fold cross validation.

The appeal of AutoML is that it reduces the expertise required for obtaining performant models using machine learning methods. Because of this there is an increasing interest in AutoML methods, as can be seen by the fact that there are currently various commercial offerings available for AutoML [28].

In Fig. 3 a schematic overview is given of the pipeline that AutoML methods try to automate. How and which components of this pipeline are automated differs per method. AdaNet (Appendix B), for example, just focusses on Neural Architecture Search (NAS), while Auto-sklearn (Section 2.2.2) does data preparation and feature engineering besides model generation and evaluation.

AutoKeras

AutoKeras [36] is an AutoML (more precisely a NAS) package built on top of Keras [9], which in turn is built on top of TensorFlow [12].

It does not follow the CASH formalisation, however the formalisation used by AutoKeras is similar. For a neural architecture search space $\mathcal{F}$, the dataset is divided into D_{train} and D_{test}. We define $\text{Cost}(A, B)$ as the evaluation metric as provided by the user (e.g. MSE, accuracy). We have:

$$f^* = \underset{f \in \mathcal{F}}{\text{argmin}}\, \text{Cost}(f(\theta^*), D_{\text{val}}) \tag{2}$$

$$\theta^* = \underset{\theta}{\text{argmin}}\, \mathcal{L}(f(\theta), D_{\text{train}}) \tag{3}$$

Here θ is the learned parameter of f.

AutoKeras explores the search space by morphing the neural architecture guided by a Bayesian optimization algorithm. The algorithm works in three steps [36]:

1. **Updating**: train and test existing models and use their results train the Bayesian optimization search,
2. **Generation**: generate the next architecture as guided by the Bayesian optimization algorithm,
3. **Observation**: Obtain the performance by training the network and test the performance on the validation set. Update the Bayesian optimization algorithm with the results.

The fact that AutoKeras does Bayesian optimization for a neural architecture search brings some challenges [36]: Bayesian optimisation techniques are normally used for learning functions in Euclidean space, which a neural architecture is not; Furthermore, transitional gradient based methods cannot be used to optimize for discrete network morphism actions; Finally, keeping the network consistent is a problem. Changes on one layer can require changes on other layers, such as shape changes of output tensors requiring changes to the shape of the input of the child layers. To tackle these issues, AutoKeras uses three key components [36]:

1. Edit-Distance Neural Network Kernel for Gaussian Processes,
2. Optimization for Tree Structured Space,
3. Graph-Level Network Morphism.

Edit-Distance Neural Network Kernel for Gaussian Processes The authors of AutoKeras propose an approximate solution to determine the edit distance between two neural networks. Approximate because finding the edit distance between two neural networks is equivalent to finding the edit distance between two graphs, an NP-hard problem [81].

Optimization for Tree Structured Space Traditional acquisition functions for Bayesian optimisation are defined on Euclidean space. These functions are not applicable for tree-structured spaces required for the morphism of networks. The authors of the AutoKeras paper propose a novel method to optimize the acquisition function on tree-structured space. They use a A*-based algorithm to exploit the most promising nodes combined with simulated annealing to balance the exploitation with exploration.

Graph-Level Network Morphism Finally the authors define the following morphism functions on a neural network:

deep Inserting a layer to the neural network.

wide Depending on the type of the previous layer to where this is used this can mean adding more filters to the previous layer if it is a convolutional layer, or making the output vector of the previous fully-connected layer longer.

add Adding an additive connection between two nodes.

concat Adding an concatenative connection between two nodes.

Auto-sklearn

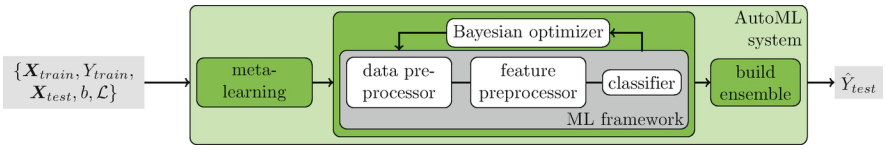


Figure 4: The general pipeline of the Auto-sklearn library. Noteworthy are the additions of the meta-learning and automated ensemble construction nodes. By Feurer *et al.* [17] (image licensed under CC BY 4.0)

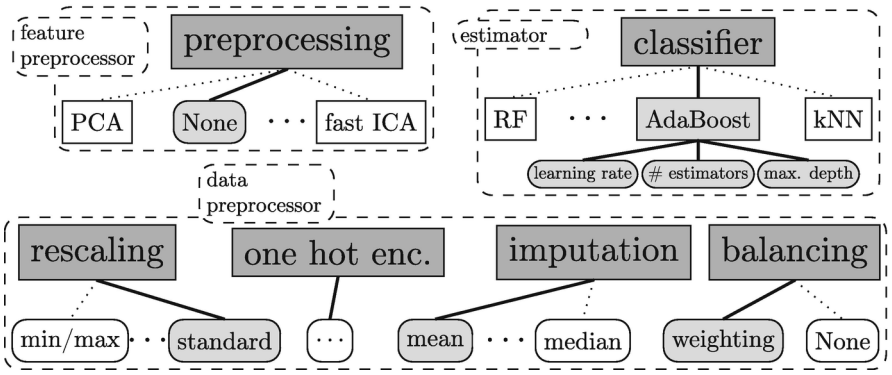


Figure 5: The configuration space of Auto-sklearn. In the three categories (feature preprocessor, estimator, and data preprocessor) we can have respectively one, one, and up to three methods used in the final pipeline. Squared boxes denote parent hyperparameters, boxes with rounded edges are leaf hyperparameters. Grey boxes denote active hyperparameters in the pipeline. By Feurer *et al.* [17] (image licensed under CC BY 4.0)

Auto-sklearn [18] is a popular AutoML package built on top of scikit-learn (sklearn) [53]. The library contains 15 classification algorithms, 14 preprocessing methods, and 4 data preprocessing methods, resulting in a total of 110 hyperparameters [18]. For the full list of the algorithms and method included, I refer you to the original Auto-sklearn paper [18]. In Fig. 4 an overview of the pipeline of Auto-sklearn is given and in Fig. 5 an overview of the configuration space is given.

Auto-sklearn follows the CASH formalisation as shown in the previous section (Eq. (1)). To optimise the formalisation Auto-sklearn uses Bayesian optimization which optimizes both the data and features preprocessors as well as the classifier selected and its hyperparameters. However, Auto-sklearn makes two additions: 1. A meta-learning step, 2. an ensemble construction step.

The meta-learning step is used to warm-start the Bayesian optimization search. Meta-learning is transferring the knowledge of older datasets to newer datasets [58]. In the case of Auto-sklearn, a large amount of pre-trained models is given corresponding to datasets which metafeatures are recorded. When Auto-sklearn is now given a new dataset, the nearest neighbours in the metalearning set compared to the new dataset are found (per default 25), and the parameters of these pre-trained models are given to the Bayesian optimization, thus aiding in the search.

For the automated ensemble construction step Auto-sklearn provides the insight that Bayesian hyperparameter optimization is wasteful. To get to the best performing model, all earlier models trained to reach the best model are discarded, even when they perform almost as well as the resulting model. Instead of discarding, Auto-sklearn stores and uses these models to construct an ensemble. For this the library uses the ensemble technique from Caruana *et al.* [7]. Ensembles of weak models are known to perform better than these individual models and are also less prone to overfitting [42, 75].

In summary, Auto-sklearn is a general purpose AutoML package, containing a large amount of methods and classifiers. Set up is such a way that the only settings the end-user has to configure are easy to understand for the data science layman (we will explore this in more detail in Section 5.2).

3 Methodology

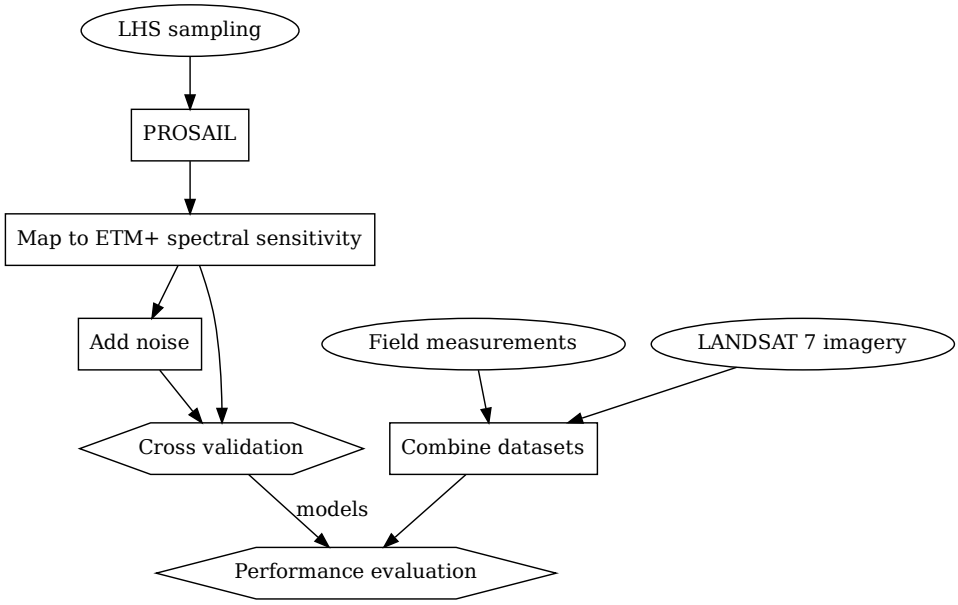


Figure 6: Flowchart showing the overall setup of the experiments done in this thesis. The circular nodes represent datasets, the box nodes represent steps in the process, and the hexagons represent steps which produce results.

In Fig. 6 a schematic overview is given of the methodology used in this research.

An artificial dataset is created, which is described in Section 3.1. We use 5-fold random permutations cross-validation (also known as Monte Carlo cross validation) [78] to train and evaluate models using the methods as described in Section 2.2. These experiments were ran using increasing training times to determine the time after which the models did not show significant improvement anymore. This was determined to be around 16000 seconds (4 hours, 26 minutes and 40 seconds) (Section 4), this training time is used as the baseline for comparisons between the methods. We added artificial inverse-combined noise of 5% (per Sá *et al.* [56]) to the dataset and trained the models with the chosen baseline setting on the dataset with artificial noise. We use these 5 models to asses their performance on the field dataset, which is described in Section 3.2.

All models were trained and evaluated on a computer running an Intel(R) Xeon(R) CPU E5-4667 v3 CPU @ 2.00GHz. Every model is trained and evaluated using one thread, with an unlimited amount of memory allowed. In practice, we have not observed the memory usage to go above 3GiB per model.

Metrics The metrics used to quantify the performance of the models are defined in Table 2. The MAE metric uses the same unit as the variable it measures, is easy to interpret and symmetrical [33]. Therefore we use the MAE metric for the general conclusions in this thesis.
The MAPE metric is scale-invariant, it uses percentages in the range $[0, \infty)$ [33]. However, it has

shortcomings: it is non symmetrical and produces produces values up to infinity for actual values close to zero [39].
The MAPE metric is used in graphs were differently scaled units are compared to each other. Here the problems of MAPE are less of a concern, and the property of scale-invariance is useful.
Finally we also calculate the RMSE metric, it is symmetrical but not linear. This metric is only used for comparison to other research. RMSE is a popular choice, however it is more sensitive to outliers, thus might be harder to interpret [77, 33]. RMSE — just like MAE — represents the error in the same range as the variable it measures.

Table 2: The metrics used for the performance comparisons in this thesis.

Name	Formula
Mean Absolute Error (MAE)	$\frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} \lvert y_i - \hat{y}_i \rvert$
Mean Absolute Percentage Error (MAPE)	$\frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} \frac{\lvert y_i - \hat{y}_i \rvert}{max(\epsilon, \lvert y_i \rvert)}$
Root Mean Square Error (RMSE)	$\sqrt{\frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2}$

Where:

n_{samples}: the amount of samples in the dataset to calculate the metric for;

y_i: the actual value;

$\hat{y}_i$: the value predicted by the model;

ϵ: an arbitrary small yet strictly positive number to ensure that, when y is equal to zero, the equation won't result in an undefined result.

3.1 Artificial data generation

We use Latin Hypercube Sampling (LHS) [50] to sample 10 000 parameter datapoints from the parameter space (Section 3.1.1). We use these datapoints to generate reflectance spectra using the PROSAIL model Section 2.1. The continuous reflectance spectra are reduced to the bands using a spectral convolution approach [56] based on the spectral response functions provided by NASA.[8]. The algorithms used for these calculations are provided in Appendix C. The AGB was calculated using Eq. (4) (from He *et al.* [27]). Thus we obtained two datasets: one multi-output dataset (LAI and C_m) and one single-output dataset (AGB).

$$\text{AGB} = \text{LAI} \cdot C_m \tag{4}$$

Artificial noise added to the dataset was generated by using the Inverse-combined noise formula (Eq. (5)) of Locherer *et al.* [47]. The noise level percentage chosen is 5%, as determined by Sá *et al.* [56]. Specifically 5% was chosen since it results in a good trade-off of bias and variance. These datasets are split using random permutations cross-validation using 5 folds, 5000 datapoints used for training and 5000 datapoints used for evaluation.

[8] `https://landsat.usgs.gov/spectral-characteristics-viewer`

$$R_{ns}(\lambda) = 1 - \{[1 - R_{sim}(\lambda)] \times [1 + \chi(0, 2\sigma(\lambda))]\} + \chi(0, \sigma(\lambda)) \quad (5)$$

Where:

$R_{ns}(\lambda)$: simulated reflectance value for band λ with added noise;

$R_{sim}(\lambda)$: simulated reflectance value for band λ;

$\chi(0, \sigma)$: Gaussian distribution with a mean value of 0 and a standard deviation of σ;

$\sigma(\lambda)$: Uncertainties within the Gaussian distribution for band λ.

3.1.1 PROSAIL parameters

In the following table we describe the parameters that are sampled using the Latin Hypercube Sampler (LHS). Per variable: the symbol, a small description, the unit and the range is given. We adapted the values used by He *et al.* [27]. We determined the ranges for the angle values from our field dataset (Section 3.2).

Table 3: The ranges for the PROSAIL parameters to be generated by the Latin Hypercube Sampler. The ⇒ indicates variables for which we are interested in estimating them.

	Description	Parameter	Unit	Range
	Leaf structure index	N		$[1.500, 1.900]$
	Chlorophyll a + b content	C_{ab}	ug/cm^2	$[15.000, 55.000]$
	Total carotenoid content	Car	ug/cm^2	10.000
	Equivalent water thickness	C_w	cm	$[0.002, 0.010]$
⇒	Dry matter content	C_m	g/cm^2	$[0.005, 0.010]$
	Brown pigments	C_{brown}		0.000
	Total anthocyanin content	C_{anth}	ug/cm^2	0.000
⇒	Leaf area index	LAI		$[0.100, 8.000]$
	Average leaf slope	LIDF_a	°	-1.000
	Leaf distribution bimodality	LIDF_b	°	0.000
	Hot spot parameter	hspot		$[0.050, 0.100]$
	Soil reflectance	p_{soil}		0.500
	Soil brightness factor	α_{soil}		0.500
	Solar zenith angle	tts	°	$[0.000, 1.000]$
	Sensor zenith angle	tto	°	$[30.000, 70.000]$
	Relative azimuth angle	phi	°	$[0.000, 340.000]$

In Fig. 7 a curve is shown that represents the generated reflectance spectra, the line is the mean value of all the spectra and the shaded area shows the 95% confidence interval.

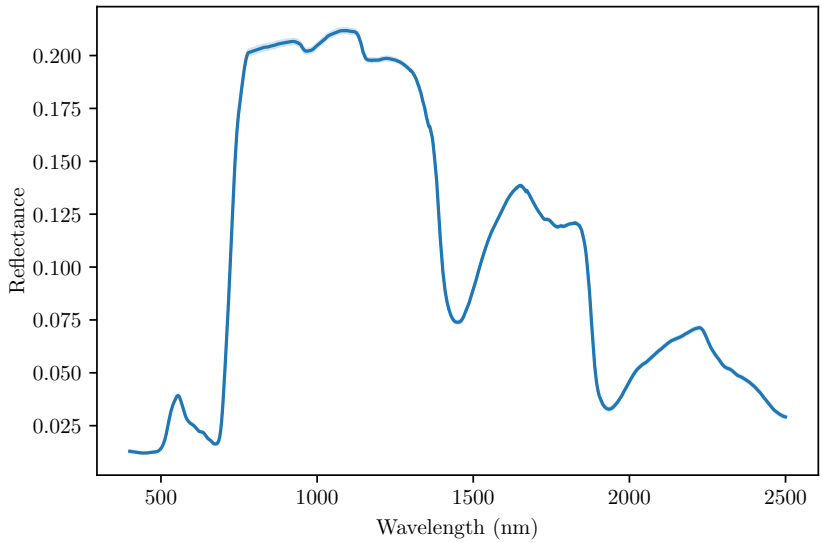


Figure 7: The reflectance spectra PROSAIL generated by PROSAIL. The shaded area shows the 95% confidence interval.

In Fig. 8 the distribution is given for the AGB parameter ($\mathrm{LAI} \cdot C_m$), visualised as a histogram. The curve shows the Kernel Density Estimate (KDE).

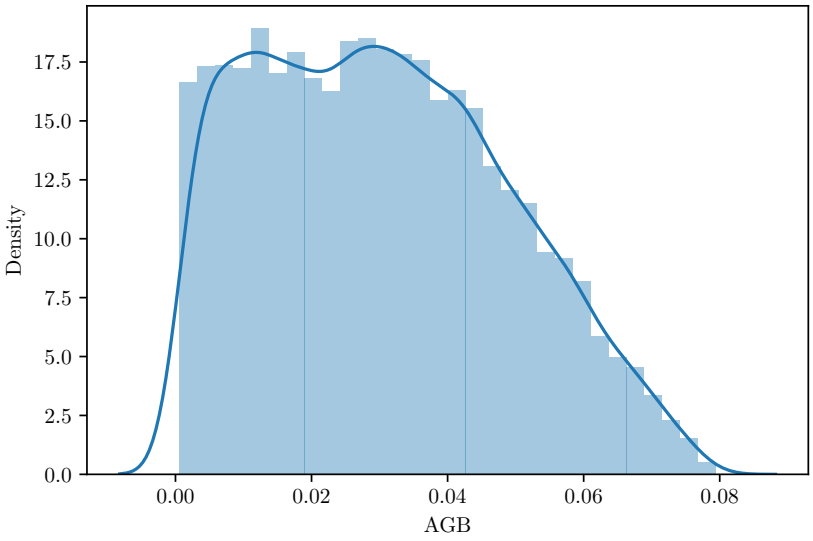


Figure 8: The distribution of the AGB parameter ($\mathrm{LAI} \cdot C_m$) visualised as a histogram. The curve shows the Kernel Density Estimate (KDE).

3.2 Field data

The field data is comprised of field measurements done in the Oostvaardersplassen combined with satellite imagery taken by the Landsat 7 mission. We will describe the study area in Section 3.2.1 and the process done to prepare the data in Section 3.2.2.

3.2.1 Study area

The study area for the field data is the Oostvaardersplassen, a nature reserve in province of Flevoland in the Netherlands built on a piece of reclaimed land. It is seen as an experiment of rewilding [48]. The Oostvaardersplassen contains both dry grasslands and wetlands [41]. Humans introduced large herbivores which graze in the dry grasslands. The wetlands is an important location for birds, used as wintering grounds. Therefore, it has also been classified as a Special Protection Area (SPA) under the European Environment Agency's Birds Directive [16].

3.2.2 Data preparation

Field data gathered by Staatsbosbeheer[9] (the Dutch governmental organisation for forestry) is used to determine the relationship between grass sward heights and the biomass of the corresponding patch. The data was gathered by transecting sections of the dry grasslands. The dataset contains both the sward heights and the biomass per patch, we used this to determine a linear relationship between the two variables for the dry grasslands using LibreOffice Calc [20] ($r^2 = 0.8007$):

$$\text{AGB } [\text{g cm}^{-2}] = -0.0120942 + 0.0147056 \cdot \text{sward height } [\text{cm}] \tag{6}$$

This dataset is not used further.

[9] `https://www.staatsbosbeheer.nl/`

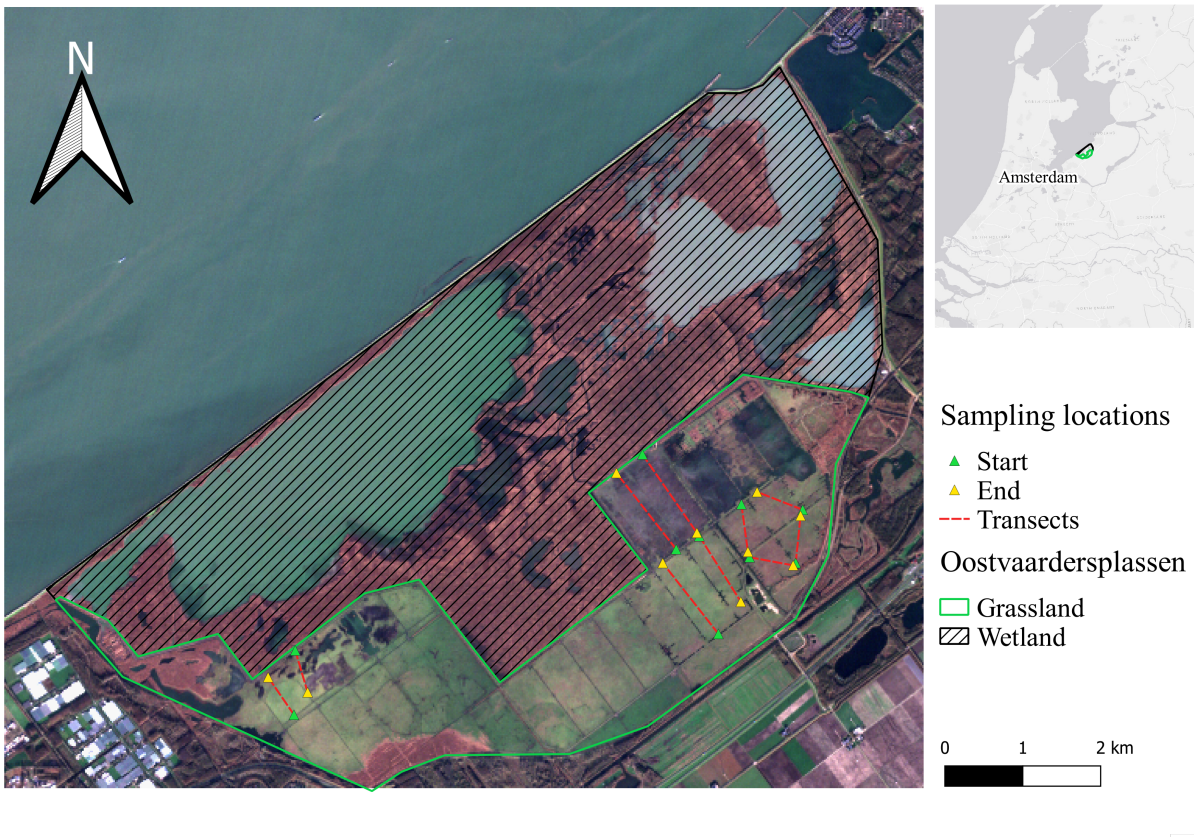


Figure 9: The transects of the dataset shown on a satellite image of the Oostvaardersplassen. Figure by Nuno De Mesquita César de Sá.

Table 5: The amount of field measurements per day in the dataset used of the Oostvaardersplassen.

Year	Month	Day	Count	RS Days difference
2013	05	07	137	-6
	06	07	170	-5
	07	08	112	-4
	08	09	171	-4
	09	12	5	10
2015	04	16	136	-5
	06	03	160	5
	07	10	174	0
2016	09	16	178	-2
	11	08	37	9
2017	06	14	175	-1
	07	25	117	6
	08	28	14	-12
	11	01	162	3

To evaluate the results of the models, we use a newer dataset, also by Staatsbosbeheer. This data again was gathered by transecting sections of the dry grasslands and measuring a sample every 50 meters. In Fig. 9 the start and end locations of these transects are shown and in Table 5 the amount of field measurements per day is shown. This dataset only contains the sward heights, so we use Eq. (6) to estimate the AGB.

The imagery for the plots are taken by the Landsat 7 mission [43] (which uses the ETM+ sensor). This sensor has eight bands, we don't use band 6 and band 8 because they are out of range for PROSAIL's output or they overlap with the other bands respectively. The ETM+ bands are (bolded are the bands that we use):

- **Band 1 Visible (0.45 - 0.52 µm)**
- **Band 2 Visible (0.52 - 0.60 µm)**
- **Band 3 Visible (0.63 - 0.69 µm)**
- **Band 4 Near-Infrared (0.77 - 0.90 µm)**
- **Band 5 Short-wave Infrared (1.55 - 1.75 µm)**
- Band 6 Thermal (10.40 - 12.50 µm)
- **Band 7 Mid-Infrared (2.08 - 2.35 µm)**
- Band 8 Panchromatic (PAN) (0.52 - 0.90 µm)

We retrieved the imagery from from Google Earth Engine [24], using the official Python library[10]. However, these images did not include the sensor angles. Therefore, we retrieved the raw angle files from the Google Cloud Landsat 7 dataset [44], which we then converted to the actual angles using the *Landsat 4-7 Angles Creation Tool* [6]. Since the angle for the sensor can differ slightly per band we took the mean value of every band per pixel. For this gathering and conversion we had to write a script which we included in the GitHub repository mentioned in Appendix C.

Finally, the field measurements and imagery were combined by taking the images of the locations that were taken the closest to the day of measurements. If there were multiple options the one with the highest quality score was chosen. To reduce the impact of noise in the imagery (since one pixel represents 30 meters on-ground) we took the average of the pixel values and measurements per plot, in Appendix C the code for this can be found in the `transform-field-data.ipynb` notebook.

3.3 Method-specific parameters

Not all methods have the same set of hyperparameters or provide the same configuration options. In this section we explain the options and ranges for the hyperparameters that we have chosen per method.

3.3.1 Traditional approach to RTM inversion

We tune the Random forests and Gaussian process regressors using random search [4], which is conveniently built-in into the scikit-optimize library [29].

[10] `https://developers.google.com/earth-engine/guides/python_install`

Random forests

Adapting from Yang and Shami [80] we optimize the following three hyperparameters:

Table 6: The hyperparameters optimised for the Random forests regressor.

Parameter	Description	Range
`max_depth`	The maximum depth of a tree in the forest.	$[5, 50]$
`min_samples_split`	Fraction of the total amount of samples that are required to split an internal node.	$[2, 11]$
`min_samples_leaf`	Fraction of the total amount of samples that are required to be at a leaf node.	$[1, 11]$

A surprising omission might be the number of estimators in the forest, we have chosen to not optimise this since we saw no improvement in performance when increasing that parameter (Fig. 18) from the default value of 100, only increasing training times. Lowering this hyperparameter also does not decrease the variance (i.e. improve any possible overfitting) [25].

Gaussian Process

We optimize the following hyperparameters:

Table 7: The hyperparameters optimised for the Gaussian process regressor.

Parameter	Description	Values
`kernel`	The covariance function to use.	{ `RBF`, `RationalQuadratic`, `Matern`, `DotProduct` }
`fixed`	Whether or not the value bounds for the kernels are fixed.	{`True`, `False`}
`n_restarts`	The number of restarts of the optimizer for finding the optimal kernels parameters.	$[10, 100]$
`alpha`	Value added to the diagonal of the kernel matrix during fitting.	$[1 \times 10^{-10}, 1]$
`scale_x`	Whether or not to MinMax scale the input values.	{`True`, `False`}
`normalize_y`	Whether or not to normalize the target values by removing the mean and scaling to unit-variance.	{`True`, `False`}

3.3.2 Automated Machine Learning

AutoKeras

AutoKeras provides the user the opportunity to tweak multiple variables, we focus on two:

`loss`: the loss metric used to evaluate the performance of the networks. This is set to Mean Squared Error (MSE) per default in AutoKeras. This default is a reliable choice for many regression problems, we therefore keep it as it is for our single-output regression. However, for multi-output regression we need to combine the metrics of the multiple regression outputs to a single metric.

Keras (and therefore AutoKeras) do this by taking the mean value of the RMSE values of the regression outputs. This introduces a problem in our case, since the outputs have completely different scales. Therefore we use Mean Absolute Percentage Error (MAPE) as the loss metric for AutoKeras.

`max_trials`: the maximum amount of neural networks to generate, train, and test. This is the main variable that the user can change to change the training time of AutoKeras, we set it to 250 trials as the baseline since this corresponds to the point after which the error does not improve in our tests with a training time bounded to 16000 as a maximum.

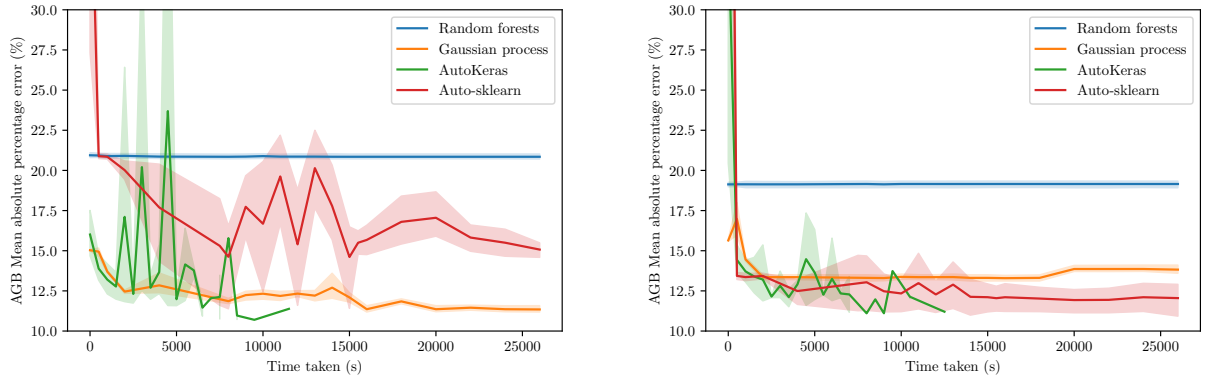
Auto-sklearn

The main variable that one can set is `time_left_for_this_task`, which indicates how many seconds Auto-sklearn can use to find and fit the pipeline it produces. This variable is easy to use since — in our testing — a higher value results in better performing models on average (Fig. 21).
The `memory_limit` parameter limits the amount of memory that can be used during the training, we set this to be unbounded.
The `n_jobs` parameter allows Auto-sklearn to be run in using multiple processor threads. We did not change this variable, meaning that only one core can be used.
Other variables were left unchanged from their default values.

4 Results

This section shows the results that directly relate to the research objectives (Section 1.2). Many intermediate results were also obtained, these are provided in Appendix A. A discussion of the results will be given in Section 5.

In the following figure we show a line plot that compares the performance of all the single-output and multi-output models on the artificial dataset by increasing time taken for training. We used this graph to determine the baseline of 16000 seconds (4 hours, 26 minutes and 40 seconds) used in this thesis. For clarity, the y-axis does not show the full range of values, since the Auto-sklearn regressor starts off with a very high MAPE value:



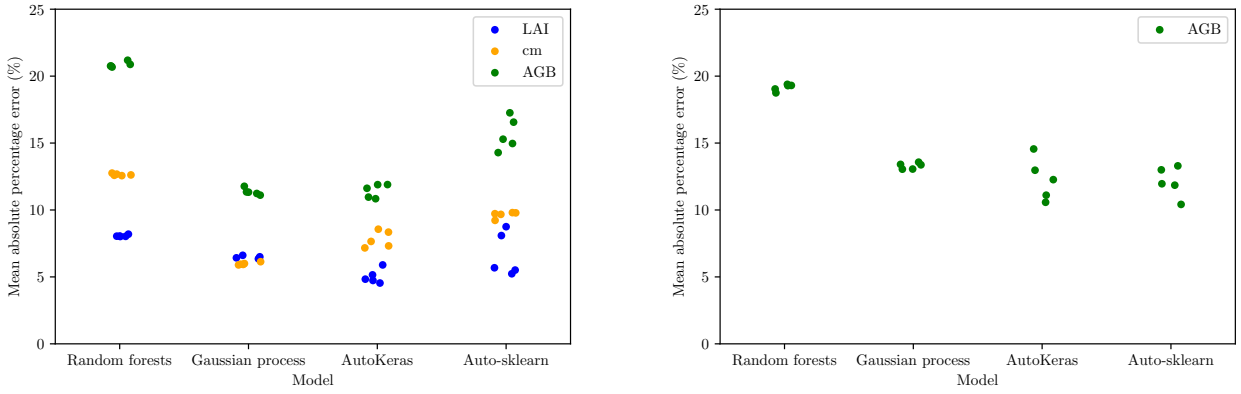
(a) Multi-output models

(b) Single-output models

Figure 10: Line plot showing the performance of the all the models on the artificial dataset expressed using the MAPE metric.

All the results shown in this section and the data used for conclusions done in this thesis were produced by the methods all being provided this amount of training time.

4.1 Artificial data



(a) Multi-output models

(b) Single-output models

Figure 11: An overview of the performance of all the models on the artificial dataset in terms of the MAPE metric visualised using a strip plot. The evaluations were done using 5-fold cross-validation. One dot represents one fold.

In Fig. 11 a strip plot visualises the performance of the models on the artificial dataset in multi-output and single-output modes. In Table 8 the mean value of the cross-validation results are shown of the best performing trained model per method using the MAE metric. In Tables 10 and 11 the results can be shown using the RMSE and MAPE metrics respectively.

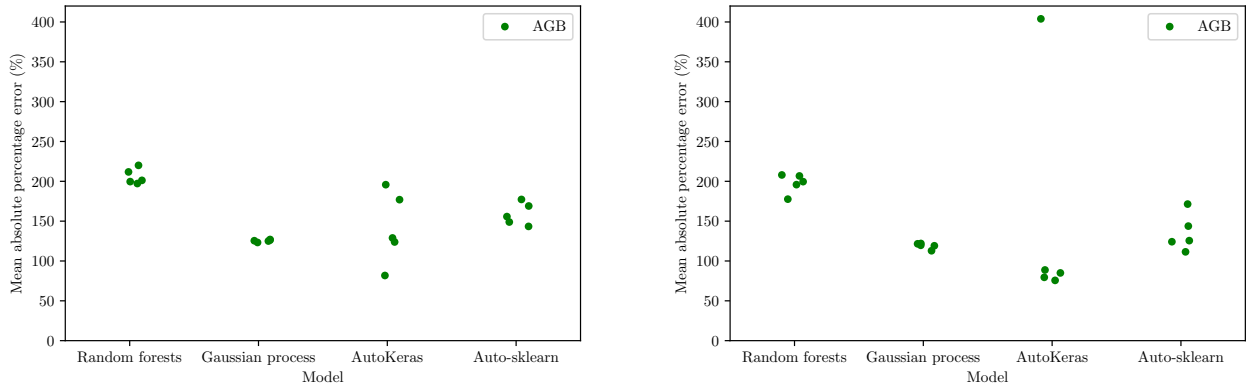
Table 8: The mean and standard deviation of the performance of the baseline models evaluated on the artificial dataset expressed in MAE. Columnwise, the best value is bolded.

Model	LAI MAE	C_m [g cm^{-2}] MAE	AGB [g cm^{-2}] MAE
Random forests	0.3356 ($\sigma = 0.0050$)	0.0009 ($\sigma = 0.0000$)	0.0057 ($\sigma = 0.0001$)
Gaussian process	0.2719 ($\sigma = 0.0054$)	**0.0004 ($\sigma = 0.0000$)**	0.0036 ($\sigma = 0.0001$)
AutoKeras	**0.2474 ($\sigma = 0.0164$)**	0.0006 ($\sigma = 0.0001$)	0.0037 ($\sigma = 0.0002$)
Auto-sklearn	0.2765 ($\sigma = 0.0246$)	0.0007 ($\sigma = 0.0000$)	0.0045 ($\sigma = 0.0002$)
Random forests single-output			0.0052 ($\sigma = 0.0001$)
Gaussian process single-output			0.0041 ($\sigma = 0.0001$)
AutoKeras single-output			0.0041 ($\sigma = 0.0005$)
Auto-sklearn single-output			**0.0034 ($\sigma = 0.0003$)**

We see in Table 8 that the single-output Auto-sklearn regressor is the best performing model for predicting the AGB value. Since the single-output regressors don't predict the LAI and C_m variables, we only have the errors metrics for these variables for the multi-output regressors. In the case for LAI the multi-ouput AutoKeras regressor has the lowest error. For C_m the best performing method is the multi-output Gaussian process regressor.

The average MAE for the AutoML methods (0.003925) was lower than the average MAE for the traditional machine learning methods (0.00465).

4.2 Field data



(a) multi-output

(b) single-output

Figure 12: An overview of the performance of the models on the field dataset in terms of MAPE visualised using a strip plot. Each dot an evaluation on the complete field dataset of one model trained on a cross-validation fold of the artificial dataset.

In Fig. 12 a strip plot visualises the performance of the models on the field dataset in multi-output and single-output modes. Note that the scale is different than that used in Fig. 11. In Table 9 the mean values are shown for the models applied on the field data. In Table 12 the same table is shown with RMSE and MAPE added.

Table 9: The mean and standard deviation of the performance of the baseline models evaluated on the field dataset expressed in the MAE metric. Columnwise, the best value is bolded.

Model	AGB [g cm^{-2}] MAE
Random forests	0.0536 ($\sigma = 0.0006$)
Gaussian process	**0.0514 ($\sigma = 0.0002$)**
AutoKeras	0.0567 ($\sigma = 0.0116$)
Auto-sklearn	0.0543 ($\sigma = 0.0015$)
Random forests single-output	0.0532 ($\sigma = 0.0005$)
Gaussian process single-output	0.0516 ($\sigma = 0.0003$)
AutoKeras single-output	0.0558 ($\sigma = 0.0059$)
Auto-sklearn single-output	0.0539 ($\sigma = 0.0022$)

The best performing model on the field dataset is the multi-output Gaussian process regressor. The single-output Gaussian process version being the runner-up. The worst performer is the multi-output AutoKeras regressor, which actually performs fairly well on the artificial dataset (Table 8).

The average MAE for the AutoML methods (0.055175) was higher than the average MAE for the traditional machine learning methods (0.05245).

5 Discussion

As we have seen we see that on the artificial dataset the best performing model is the single-output Auto-sklearn regressor. On the field data, however, it is the multi-output Gaussian process regressor. One might think that this could be that the Gaussian process regressor has a higher bias, however, as we can see in Table 8 the Gaussian process regressor still performs well even on artificial data (although is is not the best). This behaviour warrants more investigation. Furthermore, we saw that in the case of artificial data the average MAE was lower for the AutoML methods (versus the traditional methods), while on the field data the average MAE was higher for the AutoML methods. In general one can conclude that the AutoML methods have a tendency to overfit (i.e. have a high variance). This warrants further investigation, especially since this is not observed in other evaluations of AutoML methods on different datasets [73]. Since RTM inversion has unique problems (see Section 2.1.1) this problem might quire specific improvements in AutoML techniques.

It is notable that the performance of the models is way worse on the field dataset than it is on the artificial dataset; in fact, it is 14 times worse. The dataset that we used had multiple shortcomings, we will discuss some of them here. First of all, the dataset was made to research biomass directly, it was not gathered for RS research. Since the data was gathered by transecting sections, we had to deduce the coordinates from the starting point of the transects with every increment; this might result in some error. Furthermore, the resolution of the imagery is 30 meters, this combined with the scale on which the data was gathered lead to a small amount of usable data, restricting the accuracy of the results. We approximated the AGB in the field dataset by using the formula shown in Eq. (6), this is only an approximation (the r^2 being 0.8007). The height off the grass was only recorded by increments of 1 centimeter. The difference in AGB (using Eq. (6)) between a sward height of 1 and 2 centimeters is already 0.0147056 [g cm^{-2}]. One other problem

is the fact that we only had one dataset with points close to each other. This might result in non general results due to spatial correlation. If one would use multiple dataset the errors might average out better, giving a more fair representation of the model performance. In general one would expect the models to perform worse on the field dataset than on the artificial dataset due to measurement errors and noise from the sensor and environment. However, the large discrepancy in the results on the artificial dataset and the field dataset can potentially be explained due to shortcomings of the field dataset we just mentioned.

The scatterplots of the artificial dataset (Appendix A.1.1) show some general characteristics. We can see the saturation happening in all the LAI plots. This is common behaviour with estimating LAI values, the parameter becomes chaotic for little changes in the measured reflectance [3] The C_m value has a high variance in all plots. We can see a bump in all the AGB plots.

Comparison to prior research Sá *et al.* [56] compared Artificial Neural Networks, Gaussian process regressors, Multi-task Neural Networks, and Random forests regressors for the hybrid inversion of the PROSAIL RTM. The metric used was the MAPE metric. We have comparable results, although the MAPE for C_m prediction generally is higher than that of Sá *et al.* One surprising result is that the Gaussian process regressor performed surprisingly well compared to our results. In our testing the Gaussian process regressor is still the best performing one on the MAPE metric, but the performance is close to the other methods. In the case of Sá *et al.* the Gaussian process regressor achieved a MAPE close to 0, whereas we achieved 11% with the baseline configuration.

Estévez *et al.* [15] trained Gaussian process regressors in a hybrid approach using PROSAIL to predict LAI from Sentinel-2 data. The metric used was the RMSE metric. Our field dataset does not include enough information to deduce the LAI accurately. However, we can make an educated guess of the prediction performance of our methods by dividing the performance of the AGB predictions with the mean value of the C_m parameter, since RMSE uses the same units as the variable it measures. This results in the following RMSE score for predicting the LAI variable using the multi-output Gaussian process regressor: $\frac{0.0856}{0.0075} \approx 11.4133$. The RMSE in Estévez *et al.* was 0.70, which makes our result approximately 16.3048 times as large.

Gao *et al.* [22] used a non-parametric approach using the Artificial Neural Network, Support Vector Machine, k-nearest neighbour, linear and Random forests regressors. Since the mentioned paper and our research both evaluated Random forests regressors on the problem of AGB estimation for LANDSAT data, but our research used a hybrid approach and Gao *et al.* used non-parametric it might be interesting to compare. The RMSE in Gao *et al.* came to 28.4 [Mg / ha], which is 0.284 [g cm^{-2}]. This would make the RMSE in this thesis for the Random forests regressor better (our result was 0.0866 [g cm^{-2}]).

He *et al.* [27] used Lookup-table (LUT) inversion for a physically based inversion approach using the PROSAIL RTM to estimate the AGB variable from MODIS imagery. The RMSE for the AGB value He *et al.* obtained is 60.06 [g m^{-2}], which is equal to approximately 0.0060 [g cm^{-2}]. Our results for the Gaussian process regressor were equal to 0.0856 [g cm^{-2}]. Which makes our results around 14.27 times as large.

Verrelst *et al.* [69] used Lookup-table (LUT) inversion to estimate the LAI and C_{ab} variables in a physically based approach using the PROSAIL RTM applied on Sentinel-2 and -3 data. The RMSE for the LAI they obtained was 1.20. Just like two paragraphs above, we can make an educated guess that our results for the multi-output Gaussian process regressor is approximately equal to 11.4133. The RMSE for the LAI variable in Verrelst *et al.* was equal to 0.89. This makes our result

approximately 12.82 times as large.

Random forests We see in Fig. 18 and Fig. 26 (a) that the Random forests regressor has a very constant error, even when the time training increases. The reason for this might lie in the fact that the hyperparameters that are optimised for the Random forests regressor (see Section 3.3.1) don't have a lot of impact in the resulting MAPE. We can see this clearly in Fig. 27. In contrast, for the Gaussian process regressor the hyperparameters do have a lot more impact (Fig. 28). And unsurprisingly we see in Fig. 19 that the Gaussian process regressor performance improves slightly as the training time increases.

Gaussian process We see that the Gaussian process regressor performs well on both the artificial dataset (Table 8) and the field dataset (Table 9). In the case of the artificial dataset the Gaussian process regressor is not the best performing of all the methods, but it is only off by a small margin.

In Fig. 28 (c) it is interesting that we see that the RationalQuadratic[11] kernel performs the most predictable with an MAPE around 12 on average for the AGB variable. The RDF[12] kernel, on the other hand, does have a lower mean value, however the standard deviation is a lot larger. We can also see that it is useful to normalize the y-value (d).

AutoKeras In the case of the AutoKeras regressor there are some interesting characteristics to be found in Appendix A.3.3. In Appendix A.1.2 one would expect the curves to start high and decrease as the training time used increases, which is more or less what is happing with the methods. However in Fig. 20 we see that with AutoKeras this is not the case, there are multiple peaks in the MAPE when the time taken for training increases. Since AutoKeras does not provide a direct way to set the maximum allowed training time (Section 3.3.2) we had to resort to the `max_trials` hyperparameter. In Fig. 29 we see a similar figure to that of Fig. 20. Instead of putting the time taken for training on the x-axis, we put the value of the `max_trials` hyperparameter. The curve is now closer to what one might expect. This is not unexpected, whereas there is a relationship between the `max_trials` hyperparameter and the training time (Fig. 31), the training time is also influenced by how 'lucky' AutoKeras gets with itś Neural Architecture Search. Therefore, with a low value for `max_trials` the search might end up with a bad archiecture, which it cannot train to a well performing level.

In Fig. 31 we can see that there is little correlation between an increased value for the `max_trials` hyperparameter and the time taken for training. A little increase in mean training time can be seen, however the standard deviation is large regardless. Noteworthy in this figure is the fact that the multi-output regressor increases the time taken a little faster for a higher `max_trials` value than is the case for the single-output regressor.

In Fig. 32 the relationship between the `max_trials` value and the epochs used is noted. In earlier testing we found that the amount of epochs seemed to decrease when the maximum amount of allowed trials was increased. This would make sense, since with more trails a better performing neural architecture can be found which requires less epochs to trains successfully. However, as

[11] `https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.RationalQuadratic.html`

[12] `https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.RBF.html`

seen in the figure, with further testing this relationship vanished. The mean value for the epochs used stays constant with increasing `max_trials` at around 104 epochs, the standard deviation does increase however.

Auto-sklearn For Auto-sklearn the single-output regressor performs a lot better than the multi-output regressor (Table 8). Both the single-output and multi-output variants reach their optimal performance around the same time of training (Fig. 21).

In Fig. 21 (a) we see that the LAI and AGB start with a high MAPE which decreases when the training time increases. However, the C_m value is low from the start. A reason for this might be of the saturation problem described in Baret and Buis [3], the LAI is very sensitive to changes of the spectra, which makes it harder to learn the correlation. AGB being a product of LAI and C_m it would get the same issues that LAI has on it's own. However, it is noteworthy that this issue is less pronounced with all the other methods (Figs. 18 to 20).

5.1 Research questions & hypotheses

We defined one main question and two sub questions in Section 1.3 and we defined two hypotheses in Section 1.4.

For hypothesis 1 (*AutoML methods have a lower MAE than traditional machine learning methods when used as Hybrid regression methods for RTM inversion of artificial spectral data.*) we have to take a look at Table 8. Here it becomes clear that hypothesis 1 only holds for single-output regression, where Auto-sklearn is the best performing method and AutoKeras has equal performance to the Gaussian process regressor. In the case for multi-output, the Gaussian process regressor is the best performing model. The average MAE of the AutoML methods (0.003925) is lower than the average MAE of the traditional machine learning methods (0.00465). Therefore, we can conclude that hypothesis 1 does hold. Thus, we answer the first sub research question (*Do AutoML methods have a lower MAE than traditional machine learning methods when used as hybrid regression methods for RTM inversion of artificial spectral data?*) positively.

To determine whether hypothesis 2 holds, we study the results in Table 9. Here we find that hypothesis 2 does not hold, since the Gaussian process regressor is the best performing model. The average MAE of the AutoML methods (0.055175) is also higher than that of the traditional methods (0.05245). We have to answer the second sub research question (*Do AutoML methods have a lower MAE than traditional machine learning methods when used as hybrid regression methods for RTM inversion of field data?*) negatively.

The main question (*Do AutoML methods outperform traditional machine learning techniques when used as a hybrid regression model for the estimation of AGB by RTM inversion?*) cannot be answered positively in general, since the answer of sub question two is in the negative. However, in our research, AutoML methods do in fact outperform traditional methods on artificial PROSAIL data.

5.2 User experience

Previously we only compared objective measurements, such as the MAE differences in the different methods. However, a large part of the appeal of AutoML is the ability for researchers with a

minimal amount of knowledge of machine learning techniques to reliably apply machine learning to their research.

In this section we compare the different methods from a more subjective point of view. We look at the ease-of-use and functionality that doesn't fit in any single metric, but we look at the models in a case-by-case basis, looking at the pros and cons of each method.

Random forests While being a traditional machine learning method, Random forests have some characteristics that make them easier to use than other methods. The implementation in the sklearn library [53] they don't have a lot of hyperparameters. Increasing B will decrease the variance, without increasing the bias (i.e. increasing B will not lead to overfitting) [25].

The hyperparameters that one still has to set can be tuned relatively easily by using exhaustive grid search, which is conveniently built in into sklearn. Random forests are trained relatively quickly; it lends itself to be optimised efficiently using exhaustive grid search, which in itself is not an efficient search method.

Although even without optimizing the hyperparameters, the default settings for the Random forests regressor in sklearn have good performance in our evaluation (Appendix A.3.1).

One advantage that Random forests models provide is the fact that the Gini importance can be retrieved for every feature, which might provide interesting insights in one's dataset.

Gaussian processes Gaussian process, being a Bayesian method, requires more knowledge about the properties of the dataset one is training a model on. The researcher needs to choice a kernel, which will directly influence the performance of the model [14].

Gaussian processes have multiple hyperparameters to tune (Section 3.3.1). Most are not as important, and the defaults in sklearn are good enough. However, the `alpha` parameter may lead to wildly different performing models if the out-of-sample dataset has different Gaussian measurement noise shape than the training dataset. This requires some knowledge from the researcher about Gaussian process regression to understand.

Gaussian processes have two big advantages though. One is the fact that they perform pretty well on datasets with little amount of data points, which can be crucial for field measurements. Another big advantage is the fact that the uncertainty of a estimation is known [19], which is a unique aspect that can be desired in critical decision applications.

AutoKeras AutoKeras is easy to use, although the automation isn't as extensive as that of Auto-sklearn (see Section 5.2) it still is fairly automated. As said in Section 2.2.2 the library just focusses on neural architecture search.

In Section 3.3.2 we introduced the various noteworthy tweakable variables. However, in practice the main knob would be the `max_trials` variable. This enables the user to make sure AutoKeras is not going to search too extensively for the best architecture, thus taking a lot of time.

AutoKeras is usable for more tasks than the competition, while our research has been constricted to comparing the methods to numerical regression problems, AutoKeras also has support for image problems and text problems. It can even include pre-trained blocks of ResNet [26] and EfficientNet [62] in the generated architecture.

This advantage of focussing on neural architectures also has a downside, for numerical regression problems other methods than neural networks can be interesting, such as using Gaussian processes.

In contrast to Auto-sklearn, AutoKeras doesn't build an ensemble of various methods, it just builds a neural network.

For analysing the progress and performance of AutoKeras, one can use TensorBoard since AutoKeras is built on top of Keras. Furthermore, since Keras is widely used finding troubleshoot or information about analysing the network structure is easily findable online.

Auto-sklearn Auto-sklearn is very easy to use, taking care of most of the AutoML pipeline, thus leaving no hyperparameters to tune. The only parameters to really set are the time and memory limits, these parameters do not require knowledge about machine learning, just information about the machine and the patience of the researcher.

Auto-sklearn still offers flexibility, giving the researcher the options of disabling or fixing some parts of the AutoML process.

The fact that Auto-sklearn uses metalearning to accelerate the performance of learning model is also a big plus.

Another pro of Auto-sklearn is the fact that it integrates a lot of models to be added in the ensemble, which means that a researcher will be likely to obtain a performing model in the end. Even if the problem isn't easily modelled by a MLP, for example, Auto-sklearn still has a lot of different models to try, such as: Random forests, SVM, kNN, and Gaussian processes.

PipelineProfiler [52] can be used to visualize the pipeline produced by Auto-sklearn. Giving the researcher tools to obtain insights in the resulting ensemble.

6 Conclusions and Further Research

We showed that AutoML methods (AutoKeras and Auto-sklearn) did not perform better than traditional machine learning methods (Gaussian process and Random forests) in the problem of AGB inversion on Landsat 7 data using Hybrid inversion of PROSAIL RTM data. However, on artificial PROSAIL data, the AutoML methods had a lower average MAE value. On artificial data PROSAIL data the single-output Auto-sklearn did perform the best. For multi-output versions, the Gaussian process regressor performed the best. AutoKeras did not perform better than the Gaussian process regressor for both multi-output and single-output regression. On the field dataset we found that that the best performing model was Gaussian process, both in the multi-output and single-output versions. Of the two, the multi-output version had the lowest MAE. The mean MAE value of the traditional methods was lower than the mean MAE value for the AutoML methods. In Section 5.2 we made a subjective evaluation of the user experience of the differing packages. This, in combination with the objective metrics, might be useful to determine which package to use.

There were however significant shortcomings with the field dataset that we had access to for our research, which might impact the quality of our findings. Therefore, research on larger and more appropriate datasets might lead to better quality results. Our field dataset was limited to approximated AGB, evaluating the methods applies on the problem of LAI inversion might be of interest, since the LAI component was the most stable in our evaluation on artificial data. However, we could not explore whether or not this holds too when applied on satellite imagery. Per category our research tested two offerings, so more traditional methods and AutoML offerings should be evaluated. Finally, as discussed in Section 5, the fact that AutoML methods perform worse than

traditional machine learning methods warrants further research. Since this behaviour is not seen for AutoML methods in other comparisons on different types of datasets.

References

[1] Adithya Balaji and Alexander Allen. "Benchmarking Automatic Machine Learning Frameworks". In: *CoRR* abs/1808.06492 (2018). arXiv: 1808.06492. URL: http://arxiv.org/abs/1808.06492.

[2] F. Baret *et al.* "Modeled analysis of the biophysical nature of spectral shifts and comparison with information content of broad bands". In: *Remote Sensing of Environment* 41.2 (1992), pp. 133–142. ISSN: 0034-4257. DOI: https://doi.org/10.1016/0034-4257(92)90073-S. URL: https://www.sciencedirect.com/science/article/pii/003442579290073S.

[3] Frédéric Baret and Samuel Buis. "Estimating Canopy Characteristics from Remote Sensing Observations: Review of Methods and Associated Problems". In: *Advances in Land Remote Sensing: System, Modeling, Inversion and Application*. Ed. by Shunlin Liang. Dordrecht: Springer Netherlands, 2008, pp. 173–201. ISBN: 978-1-4020-6450-0. DOI: 10.1007/978-1-4020-6450-0_7. URL: https://doi.org/10.1007/978-1-4020-6450-0_7.

[4] James Bergstra and Yoshua Bengio. "Random Search for Hyper-Parameter Optimization". In: *Journal of Machine Learning Research* 13.10 (2012), pp. 281–305. URL: http://jmlr.org/papers/v13/bergstra12a.html.

[5] Leo Breiman. In: *Machine Learning* 45.1 (2001), pp. 5–32. DOI: 10.1023/a:1010933404324. URL: https://doi.org/10.1023/a:1010933404324.

[6] *Calibration & Validation*. URL: https://www.usgs.gov/core-science-systems/nli/landsat/solar-illumination-and-sensor-viewing-angle-coefficient-files?qt-science_support_page_related_con=1#qt-science_support_page_related_con.

[7] Rich Caruana *et al.* "Ensemble Selection from Libraries of Models". In: *Proceedings of the Twenty-First International Conference on Machine Learning*. ICML '04. Banff, Alberta, Canada: Association for Computing Machinery, 2004, p. 18. ISBN: 1581138385. DOI: 10.1145/1015330.1015432. URL: https://doi.org/10.1145/1015330.1015432.

[8] Jean-Paul Chilès and Nicolas Desassis. "Fifty Years of Kriging". In: *Handbook of Mathematical Geosciences: Fifty Years of IAMG*. Ed. by B.S. Daya Sagar, Qiuming Cheng and Frits Agterberg. Cham: Springer International Publishing, 2018, pp. 589–612. ISBN: 978-3-319-78999-6. DOI: 10.1007/978-3-319-78999-6_29. URL: https://doi.org/10.1007/978-3-319-78999-6%5C%5F29.

[9] François Chollet *et al.* *Keras*. https://keras.io. 2015.

[10] *Classification and regression trees*. eng. The Wadsworth statistics/probability series 840271069. Belmont, Calif.: Wadsworth International Group, 1984. ISBN: 0534980538.

[11] Nicholas C. Coops and Thoreau Rory Tooke. "Introduction to Remote Sensing". In: *Learning Landscape Ecology: A Practical Guide to Concepts and Techniques*. Ed. by Sarah E. Gergel and Monica G. Turner. New York, NY: Springer New York, 2017, pp. 3–19. ISBN: 978-1-4939-6374-4. DOI: 10.1007/978-1-4939-6374-4_1. URL: https://doi.org/10.1007/978-1-4939-6374-4_1.

[12] TensorFlow Developers. *TensorFlow*. Version v2.4.3. Specific TensorFlow versions can be found in the "Versions" list on the right side of this page.¡br¿See the full list of authors ¡a href="htt ps://github.com/tensorflow/tensorflow/graphs/contr ibutors"¿on GitHub¡/a¿. Aug. 2021. DOI: 10.5281/zenodo.5189249. URL: https://doi.org/10.5281/zenodo.5189249.

[13] Katja Dörnhöfer and Natascha Oppelt. "Remote sensing for lake research and monitoring – Recent advances". In: *Ecological Indicators* 64 (2016), pp. 105–122. ISSN: 1470-160X. DOI: https://doi.org/10.1016/j.ecolind.2015.12.009. URL: https://www.sciencedirect.com/science/article/pii/S1470160X15007141.

[14] David Duvenaud. URL: https://www.cs.toronto.edu/~duvenaud/cookbook/.

[15] José Estévez *et al.* "Gaussian processes retrieval of LAI from Sentinel-2 top-of-atmosphere radiance data". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 167 (Sept. 2020), pp. 289–304. ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2020.07.004. URL: http://dx.doi.org/10.1016/j.isprsjprs.2020.07.004.

[16] *EUNIS - Site factsheet for Oostvaardersplassen*. URL: https://eunis.eea.europa.eu/sites/NL9802054.

[17] Matthias Feurer *et al.* "Auto-sklearn: Efficient and Robust Automated Machine Learning". en. In: *Automated Machine Learning*. Ed. by Frank Hutter, Lars Kotthoff and Joaquin Vanschoren. Cham: Springer International Publishing, 2019, pp. 113–134. ISBN: 9783030053178 9783030053185. DOI: 10.1007/978-3-030-05318-5_6. URL: http://link.springer.com/10.1007/978-3-030-05318-5%5C%5F6.

[18] Matthias Feurer *et al.* "Efficient and Robust Automated Machine Learning". In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes *et al.* Curran Associates, Inc., 2015, pp. 2962–2970. URL: http://papers.nips.cc/paper/5872-efficient-and-robust-automated-machine-learning.pdf.

[19] Christian Fiedler, Carsten W. Scherer and Sebastian Trimpe. "Practical and Rigorous Uncertainty Bounds for Gaussian Process Regression". In: *CoRR* abs/2105.02796 (2021). arXiv: 2105.02796. URL: https://arxiv.org/abs/2105.02796.

[20] The Document Foundation. *LibreOffice Calc*. URL: https://www.libreoffice.org/discover/calc/.

[21] F Fromard *et al.* "Structure, above-Ground Biomass and Dynamics of Mangrove Ecosystems: New Data from French Guiana". eng. In: *Oecologia* 115.1/2 (1998), pp. 39–53. ISSN: 0029-8549.

[22] Yukun Gao *et al.* "Comparative Analysis of Modeling Algorithms for Forest Aboveground Biomass Estimation in a Subtropical Region". en. In: *Remote Sensing* 10.4 (Apr. 2018), p. 627. ISSN: 2072-4292. DOI: 10.3390/rs10040627. URL: http://www.mdpi.com/2072-4292/10/4/627.

[23] 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). ISSN: 2072-4292. DOI: 10.3390/rs8020119. URL: https://www.mdpi.com/2072-4292/8/2/119.

[24] Noel Gorelick *et al.* "Google Earth Engine: Planetary-scale geospatial analysis for everyone". In: *Remote Sensing of Environment* (2017). DOI: 10.1016/j.rse.2017.06.031. URL: https://doi.org/10.1016/j.rse.2017.06.031.

[25] Trevor Hastie, Robert Tibshirani and Jerome Friedman. "Random Forests". In: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer New York, 2009, pp. 587–604. ISBN: 978-0-387-84858-7. DOI: 10.1007/978-0-387-84858-7_15. URL: https://doi.org/10.1007/978-0-387-84858-7%5C%5F15.

[26] Kaiming He *et al.* "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[27] Li He *et al.* "Retrieval of Grassland Aboveground Biomass through Inversion of the PROSAIL Model with MODIS Imagery". en. In: *Remote Sensing* 11.13 (July 2019), p. 1597. ISSN: 2072-4292. DOI: 10.3390/rs11131597. URL: https://www.mdpi.com/2072-4292/11/13/1597.

[28] Xin He, Kaiyong Zhao and Xiaowen Chu. "AutoML: A Survey of the State-of-the-Art". en. In: *Knowledge-Based Systems* 212 (Jan. 2021). arXiv: 1908.00709, p. 106622. ISSN: 09507051. DOI: 10.1016/j.knosys.2020.106622. URL: http://arxiv.org/abs/1908.00709.

[29] Tim Head *et al. scikit-optimize/scikit-optimize*. Version v0.8.1. Sept. 2020. DOI: 10.5281/zenodo.4014775. URL: https://doi.org/10.5281/zenodo.4014775.

[30] Tin Kam Ho. "A Data Complexity Analysis of Comparative Advantages of Decision Forest Constructors". In: *Pattern Analysis & Applications* 5.2 (June 2002), pp. 102–112. DOI: 10.1007/s100440200009. URL: https://doi.org/10.1007/s100440200009.

[31] Tin Kam Ho. "Random decision forests". In: *Proceedings of 3rd International Conference on Document Analysis and Recognition*. Vol. 1. 1995, 278–282 vol.1. DOI: 10.1109/ICDAR.1995.598994.

[32] J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[33] Rob J. Hyndman and Anne B. Koehler. "Another look at measures of forecast accuracy". In: *International Journal of Forecasting* 22.4 (2006), pp. 679–688. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2006.03.001. URL: https://www.sciencedirect.com/science/article/pii/S0169207006000239.

[34] S. Jacquemoud and F. Baret. "PROSPECT: A model of leaf optical properties spectra". In: *Remote Sensing of Environment* 34.2 (1990), pp. 75–91. ISSN: 0034-4257. DOI: https://doi.org/10.1016/0034-4257(90)90100-Z. URL: https://www.sciencedirect.com/science/article/pii/003442579090100Z.

[35] Stéphane Jacquemoud *et al.* "PROSPECT+SAIL models: A review of use for vegetation characterization". In: *Remote Sensing of Environment* 113 (2009). Imaging Spectroscopy Special Issue, S56–S66. ISSN: 0034-4257. DOI: https://doi.org/10.1016/j.rse.2008.01.026. URL: https://www.sciencedirect.com/science/article/pii/S0034425709000765.

[36] Haifeng Jin, Qingquan Song and Xia Hu. "Auto-Keras: An Efficient Neural Architecture Search System". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM. 2019, pp. 1946–1956.

[37] Aaron G Kamoske *et al.* "Leaf traits and canopy structure together explain canopy functional diversity: an airborne remote sensing approach". eng. In: *Ecological applications* 31.2 (2021), e02230–n/a. ISSN: 1051-0761.

[38] Teja Kattenborn. "Linking Canopy Reflectance and Plant Functioning through Radiative Transfer Models". PhD thesis. Karlsruher Institut für Technologie (KIT), 2019. 135 pp. DOI: 10.5445/IR/1000089168.

[39] Sungil Kim and Heeyoung Kim. "A new metric of absolute percentage error for intermittent demand forecasts". In: *International Journal of Forecasting* 32.3 (2016), pp. 669–679. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2015.12.003. URL: https://www.sciencedirect.com/science/article/pii/S0169207016000121.

[40] Jason M. Klusowski. *Analyzing CART*. 2020. arXiv: 1906.10086 [stat.ML].

[41] N. Kolen *et al.* "Vegetatie, begrazing en vogels in een zoetwatermoeras : monitoringsprogramma Oostvaardersplassen 1999/2000". Dutch. In: *Rijkswaterstaat Rapportendatabank* (2001).

[42] Alexandre Lacoste *et al.* "Agnostic Bayesian Learning of Ensembles". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 1. Bejing, China: PMLR, 22–24 Jun 2014, pp. 611–619. URL: https://proceedings.mlr.press/v32/lacoste14.html.

[43] *Landsat 7*. URL: https://www.usgs.gov/core-science-systems/nli/landsat/landsat-7.

[44] *Landsat data — Cloud Storage — Google Cloud*. URL: https://cloud.google.com/storage/docs/public-datasets/landsat.

[45] Trang T Le, Weixuan Fu and Jason H Moore. "Scaling tree-based automated machine learning to biomedical big data with a feature set selector". In: *Bioinformatics* 36.1 (2020), pp. 250–256.

[46] Erin LeDell and Sebastien Poirier. "H2O AutoML: Scalable Automatic Machine Learning". In: *7th ICML Workshop on Automated Machine Learning (AutoML)* (July 2020). URL: https://www.automl.org/wp-content/uploads/2020/07/AutoML_2020_paper_61.pdf.

[47] Matthias Locherer *et al.* "Retrieval of Seasonal Leaf Area Index from Simulated EnMAP Data through Optimized LUT-Based Inversion of the PROSAIL Model". In: *Remote Sensing* 7.8 (2015), pp. 10321–10346. ISSN: 2072-4292. DOI: 10.3390/rs70810321. URL: https://www.mdpi.com/2072-4292/7/8/10321.

[48] J. Lorimer and C.P.G. Driessen. "Experiments with the wild at the Oostvaardersplassen". English. In: *Ecos* 35.3/4 (2014), pp. 44–52. ISSN: 0143-9073.

[49] Vitor S. Martins *et al.* "Seasonal and interannual assessment of cloud cover and atmospheric constituents across the Amazon (2000–2015): Insights for remote sensing and climate analysis". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 145 (2018). SI: Latin America Issue, pp. 309–327. ISSN: 0924-2716. DOI: https://doi.org/10.1016/j.isprsjprs.2018.05.013. URL: https://www.sciencedirect.com/science/article/pii/S0924271618301461.

[50] M. D. McKay, R. J. Beckman and W. J. Conover. "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code". In: *Technometrics* 21.2 (1979), pp. 239–245. ISSN: 00401706. URL: http://www.jstor.org/stable/1268522.

[51] A. O'Hagan. "Curve Fitting and Optimal Design for Prediction". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 40.1 (1978), pp. 1–42. ISSN: 00359246. URL: http://www.jstor.org/stable/2984861.

[52] Jorge Piazentin Ono *et al.* *PipelineProfiler: A Visual Analytics Tool for the Exploration of AutoML Pipelines*. 2020. arXiv: 2005.00160 [cs.HC].

[53] F. Pedregosa *et al.* "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[54] Shaun Quegan *et al.* "The European Space Agency BIOMASS mission: Measuring forest above-ground biomass from space". In: *Remote Sensing of Environment* 227 (2019), pp. 44–60. ISSN: 0034-4257. DOI: https://doi.org/10.1016/j.rse.2019.03.032. URL: https://www.sciencedirect.com/science/article/pii/S0034425719301233.

[55] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. OCLC: ocm61285753. Cambridge, Mass: MIT Press, 2006. ISBN: 978-0-262-18253-9.

[56] 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). ISSN: 2072-4292. DOI: 10.3390/rs13040648. URL: https://www.mdpi.com/2072-4292/13/4/648.

[57] G. Arturo Sánchez-Azofeifa *et al.* "Differences in leaf traits, leaf internal structure, and spectral reflectance between two communities of lianas and trees: Implications for remote sensing in tropical environments". eng. In: *Remote sensing of environment* 113.10 (2009), pp. 2076–2088. ISSN: 0034-4257.

[58] T. Schaul and J. Schmidhuber. "Metalearning". In: *Scholarpedia* 5.6 (2010). revision #91489, p. 4650. DOI: 10.4249/scholarpedia.4650.

[59] Román A. Serrago *et al.* "Foliar diseases affect the eco-physiological attributes linked with yield and biomass in wheat (Triticum aestivum L.)" In: *European Journal of Agronomy* 31.4 (2009), pp. 195–203. ISSN: 1161-0301. DOI: https://doi.org/10.1016/j.eja.2009.06.002. URL: https://www.sciencedirect.com/science/article/pii/S1161030109000501.

[60] Ge Shen *et al.* "Monitoring wind farms occupying grasslands based on remote-sensing data from China's GF-2 HD satellite—A case study of Jiuquan city, Gansu province, China". In: *Resources, Conservation and Recycling* 121 (2017). Environmental Challenges and Potential Solutions of China's Power Sector, pp. 128–136. ISSN: 0921-3449. DOI: https://doi.org/10.1016/j.resconrec.2016.06.026. URL: https://www.sciencedirect.com/science/article/pii/S092134491630163X.

[61] Daniel Heestermans Svendsen *et al.* "Inference over radiative transfer models using variational and expectation maximization methods". In: *Machine Learning* (June 2021). DOI: 10.1007/s10994-021-05999-4. URL: https://doi.org/10.1007/s10994-021-05999-4.

[62] Mingxing Tan and Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *CoRR* abs/1905.11946 (2019). arXiv: 1905.11946. URL: http://arxiv.org/abs/1905.11946.

[63] Chris Thornton *et al.* "Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms". en. In: *arXiv:1208.3719 [cs]* (Mar. 2013). arXiv: 1208.3719. URL: http://arxiv.org/abs/1208.3719.

[64] Anh Truong *et al.* "Towards Automated Machine Learning: Evaluation and Comparison of AutoML Approaches and Tools". In: *CoRR* abs/1908.05557 (2019). arXiv: 1908.05557. URL: http://arxiv.org/abs/1908.05557.

[65] Frans WM Vera. "Large-scale nature development–The Oostvaardersplassen". In: *British Wildlife* 20.5 (2009), p. 28. URL: http://media.longnow.org/files/2/REVIVE/BritishWildlifeVera.pdf.

[66] W. 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. ISSN: 0034-4257. DOI: https://doi.org/10.1016/0034-4257(84)90057-9. URL: https://www.sciencedirect.com/science/article/pii/0034425784900579.

[67] Jochem Verrelst *et al.* "Active Learning Methods for Efficient Hybrid Biophysical Variable Retrieval". In: *IEEE Geoscience and Remote Sensing Letters* 13.7 (2016), pp. 1012–1016. DOI: 10.1109/LGRS.2016.2560799.

[68] Jochem Verrelst *et al.* "Emulation of radiative transfer models (RTMs): new opportunities for spectroscopy data processing". In: (Apr. 2017). DOI: 10.13140/RG.2.2.32771.09764.

[69] Jochem Verrelst *et al.* "Optimizing LUT-Based RTM Inversion for Semiautomatic Mapping of Crop Biophysical Parameters from Sentinel-2 and -3 Data: Role of Cost Functions". In: *IEEE Transactions on Geoscience and Remote Sensing* 52.1 (2014), pp. 257–269. DOI: 10.1109/TGRS.2013.2238242.

[70] Jochem Verrelst *et al.* "Quantifying Vegetation Biophysical Variables from Imaging Spectroscopy Data: A Review on Retrieval Methods". In: *Surveys in Geophysics* 40.3 (June 2018), pp. 589–629. ISSN: 1573-0956. DOI: 10.1007/s10712-018-9478-y. URL: http://dx.doi.org/10.1007/s10712-018-9478-y.

[71] Jie Wang. *An Intuitive Tutorial to Gaussian Processes Regression*. 2021. arXiv: 2009.10862 [stat.ML].

[72] Suizi Wang *et al.* "Effects of Grazing Exclusion on Biomass Growth and Species Diversity among Various Grassland Types of the Tibetan Plateau". In: *Sustainability* 11.6 (2019). ISSN: 2071-1050. DOI: 10.3390/su11061705. URL: https://www.mdpi.com/2071-1050/11/6/1705.

[73] Jonathan Waring, Charlotta Lindvall and Renato Umeton. "Automated machine learning: Review of the state-of-the-art and opportunities for healthcare". In: *Artificial Intelligence in Medicine* 104 (2020), p. 101822. ISSN: 0933-3657. DOI: https://doi.org/10.1016/j.artmed.2020.101822. URL: https://www.sciencedirect.com/science/article/pii/S0933365719310437.

[74] Michael L. Waskom. "seaborn: statistical data visualization". In: *Journal of Open Source Software* 6.60 (2021), p. 3021. DOI: 10.21105/joss.03021. URL: https://doi.org/10.21105/joss.03021.

[75] Charles Weill *et al.* *AdaNet: A Scalable and Flexible Framework for Automatically Learning Ensembles*. 2019. arXiv: 1905.00080 [cs.LG].

[76] Toby C Wilkinson and Anja Slawisch. "An agro-pastoral palimpsest: new insights into the historical rural economy of the Milesian peninsula from aerial and remote-sensing imagery". eng. In: *Anatolian studies* 70 (2020), pp. 181–206. ISSN: 0066-1546.

[77] Cort J Willmott and Kenji Matsuura. "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance". In: *Climate research* 30.1 (2005), pp. 79–82.

[78] Qing-Song Xu and Yi-Zeng Liang. "Monte Carlo cross validation". In: *Chemometrics and Intelligent Laboratory Systems* 56.1 (2001), pp. 1–11. ISSN: 0169-7439. DOI: https://doi.org/10.1016/S0169-7439(00)00122-2. URL: https://www.sciencedirect.com/science/article/pii/S0169743900001222.

[79] Yiming Xu *et al.* "Estimating soil total nitrogen in smallholder farm settings using remote sensing spectral indices and regression kriging". In: *CATENA* 163 (2018), pp. 111–122. ISSN: 0341-8162. DOI: https://doi.org/10.1016/j.catena.2017.12.011. URL: https://www.sciencedirect.com/science/article/pii/S0341816217304071.

[80] Li Yang and Abdallah Shami. "On hyperparameter optimization of machine learning algorithms: Theory and practice". In: *Neurocomputing* 415 (2020), pp. 295–316. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2020.07.061. URL: https://www.sciencedirect.com/science/article/pii/S0925231220311693.

[81] Zhiping Zeng *et al.* "Comparing Stars: On Approximating Graph Edit Distance". In: *Proc. VLDB Endow.* 2.1 (Aug. 2009), pp. 25–36. ISSN: 2150-8097. DOI: 10.14778/1687627.1687631. URL: https://doi.org/10.14778/1687627.1687631.

[82] R. Zurita-Milla, V.C.E. Laurent and J.A.E. 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). Special Issue on "Advances in remote sensing of vegetation function and traits", pp. 7–18. ISSN: 0303-2434. DOI: https://doi.org/10.1016/j.jag.2015.02.003. URL: https://www.sciencedirect.com/science/article/pii/S0303243415000355.

A Auxiliary results

More graphs and figures were produced than were shown in Section 4, because they were not helpful to form answers to the research questions (Section 1.3) or hypotheses (Section 1.4). In this appendix these figures are shown, which might be of interest.

A.1 Artificial data

Table 10: The mean and standard deviation of the performance of the baseline models evaluated on the artificial dataset expressed in RMSE.

Model	LAI RMSE	C_m [g cm^{-2}] RMSE	AGB [g cm^{-2}] RMSE
Random forests	0.5028 ($\sigma = 0.0061$)	0.0011 ($\sigma = 0.0000$)	0.0079 ($\sigma = 0.0001$)
Gaussian process	**0.4450 ($\sigma = 0.0078$)**	**0.0006 ($\sigma = 0.0000$)**	0.0062 ($\sigma = 0.0001$)
AutoKeras	0.4702 ($\sigma = 0.0238$)	0.0008 ($\sigma = 0.0001$)	0.0066 ($\sigma = 0.0004$)
Auto-sklearn	0.4752 ($\sigma = 0.0291$)	0.0009 ($\sigma = 0.0000$)	0.0070 ($\sigma = 0.0003$)
Random forests single-output			0.0074 ($\sigma = 0.0001$)
Gaussian process single-output			0.0067 ($\sigma = 0.0001$)
AutoKeras single-output			0.0072 ($\sigma = 0.0006$)
Auto-sklearn single-output			**0.0059 ($\sigma = 0.0002$)**

Table 11: The mean and standard deviation of the performance of the baseline models evaluated on the artificial dataset expressed in MAPE.

Model	LAI MAPE	C_m [g cm^{-2}] MAPE	AGB [g cm^{-2}] MAPE
Random forests	8.0674 ($\sigma = 0.0717$)	12.6426 ($\sigma = 0.0765$)	20.8497 ($\sigma = 0.2018$)
Gaussian process	6.4602 ($\sigma = 0.1016$)	**5.9823 ($\sigma = 0.0936$)**	**11.3588 ($\sigma = 0.2492$)**
AutoKeras	**5.0312 ($\sigma = 0.5311$)**	7.8093 ($\sigma = 0.6201$)	11.4430 ($\sigma = 0.5086$)
Auto-sklearn	6.6538 ($\sigma = 1.6371$)	9.6413 ($\sigma = 0.2390$)	15.6727 ($\sigma = 1.2124$)
Random forests single-output			19.1547 ($\sigma = 0.2612$)
Gaussian process single-output			13.2914 ($\sigma = 0.2307$)
AutoKeras single-output			12.2952 ($\sigma = 1.5774$)
Auto-sklearn single-output			12.1048 ($\sigma = 1.1353$)

In the following figure the MAPE is shown for every run of the AutoKeras and Auto-sklearn regressors with differing time taken for training:

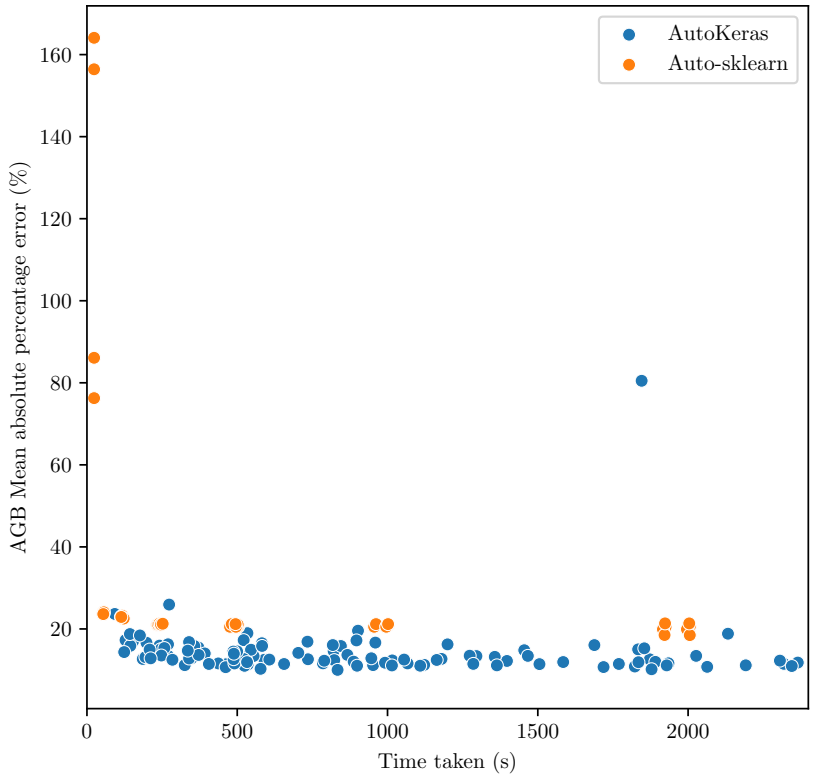
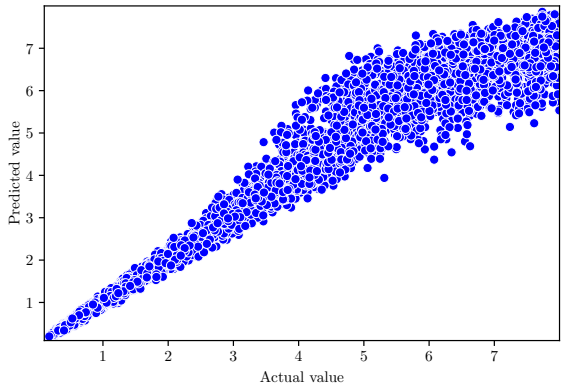


Figure 13: Comparison between performance and training time of AutoKeras and Auto-sklearn. Each dot represents one cross-validation fold evaluation result.

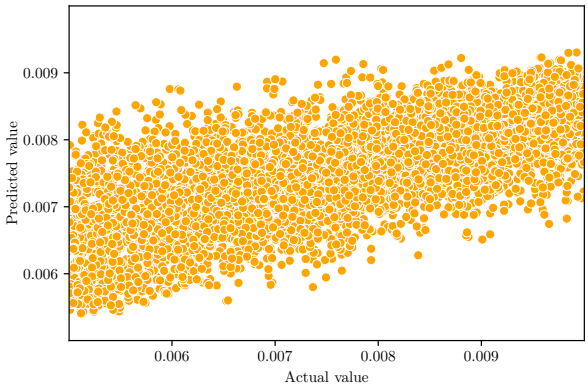
Of note in the above figure is that Auto-sklearn starts off with a very high error, decreasing rapidly. At around 2 minutes of training time the error stabilises, never reaching quite the level of performance that AutoKeras has.

A.1.1 Scatterplots

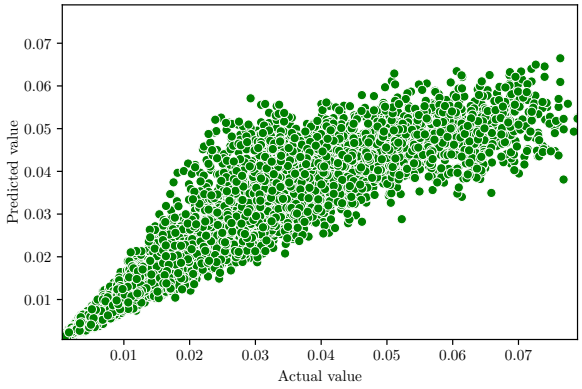
In this section scatterplots are presented hat show the performance for both the multi-output and single-output versions of all models. The models are evaluated on the artificial dataset and scatterplots are shown for every variable. All models were trained for the training time set as the baseline in Section 4. The x-axis relates to the actual value, while the y-axis relates to the estimated value.



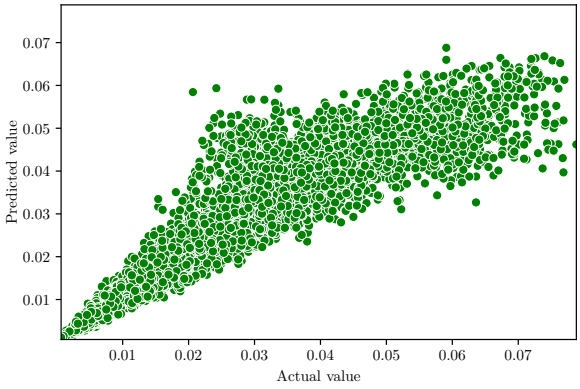
(a) LAI multi-output



(b) C_m multi-output

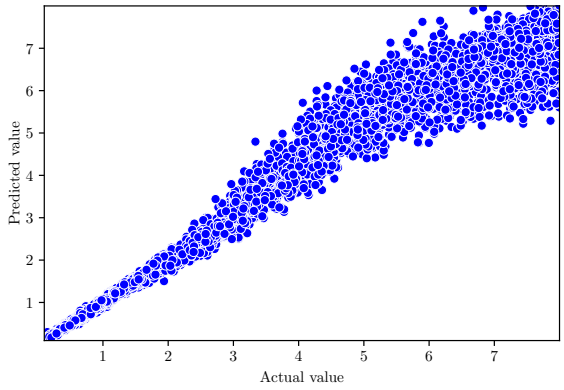


(c) AGB multi-output

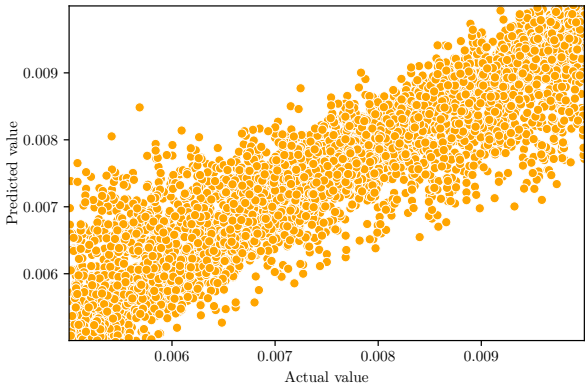


(d) AGB single-output

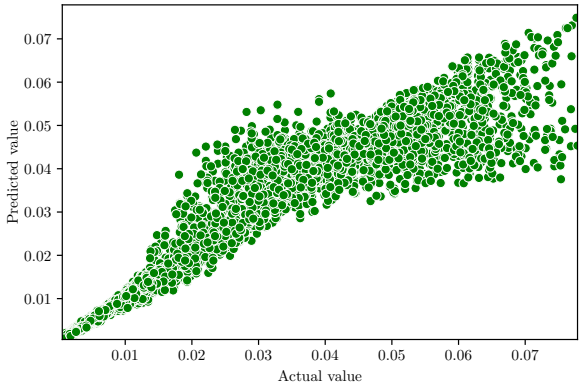
Figure 14: Random forests



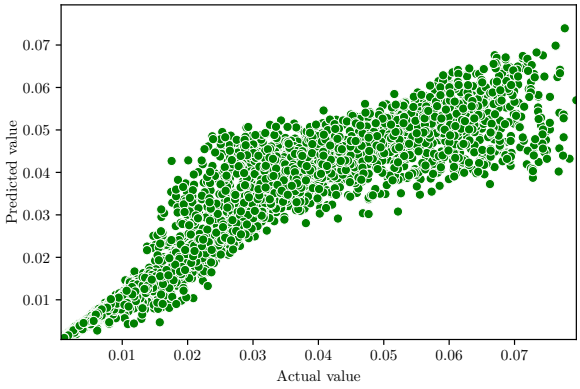
(a) LAI multi-output



(b) C_m multi-output

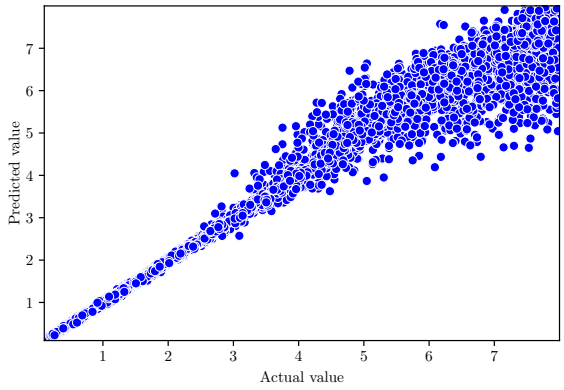


(c) AGB multi-output

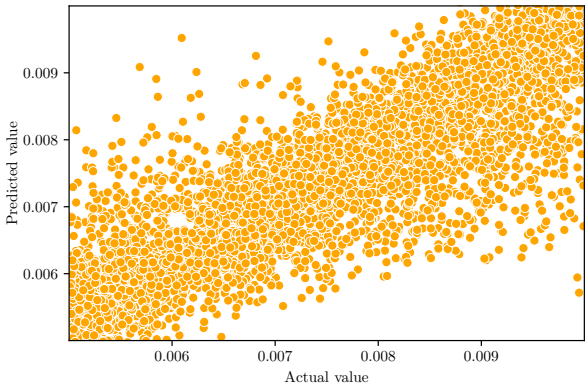


(d) AGB single-output

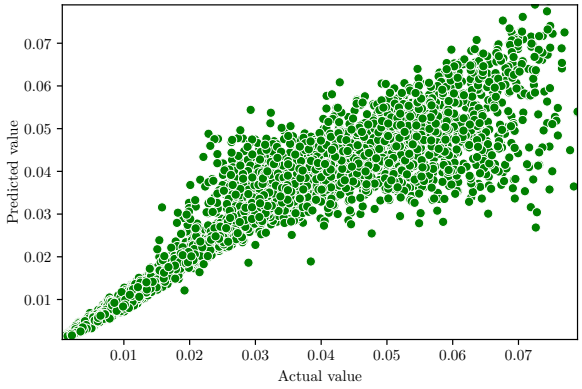
Figure 15: Gaussian process



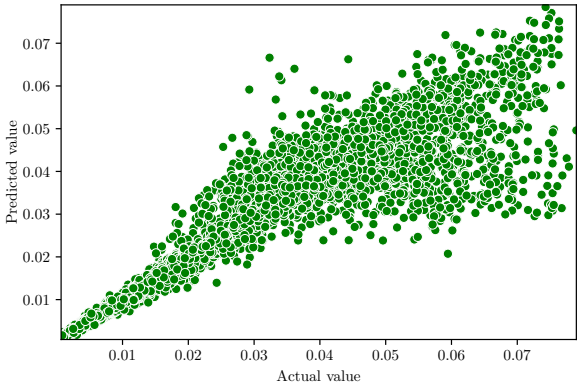
(a) LAI multi-output



(b) C_m multi-output

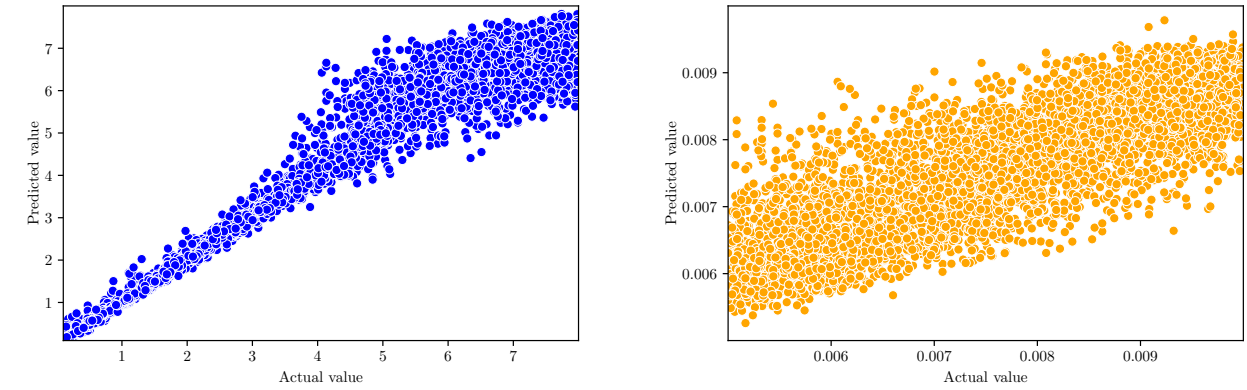


(c) AGB multi-output



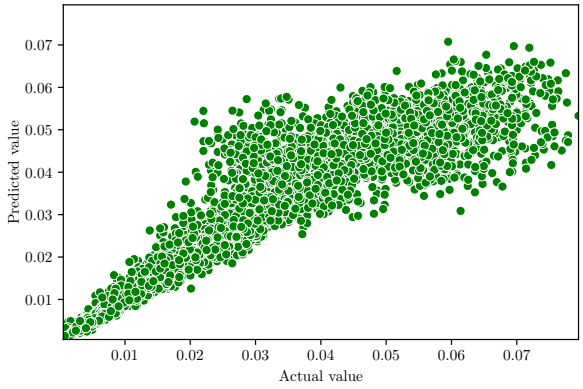
(d) AGB single-output

Figure 16: AutoKeras

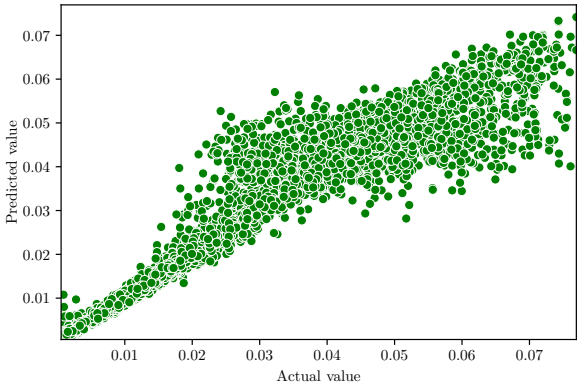


(a) LAI multi-output

(b) C_m multi-output



(c) AGB multi-output

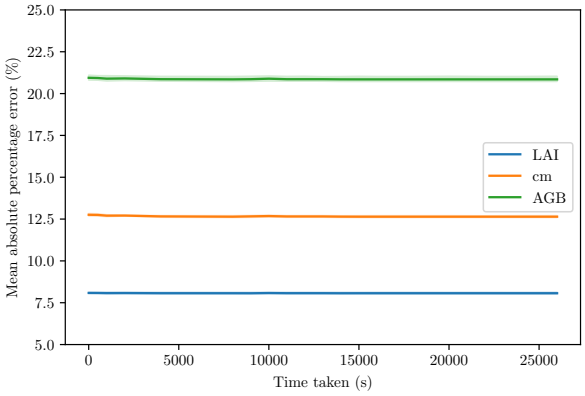


(d) AGB single-output

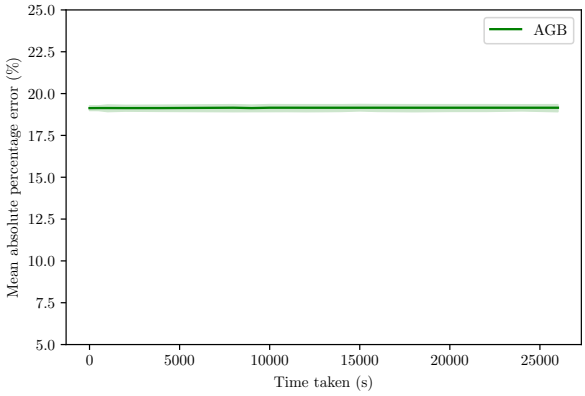
Figure 17: Auto-sklearn

A.1.2 Relationship between training time and performance

For every model, the figures in this section show the relationship between the time taken for training the model and the performance of the model expressed using the MAPE (Mean absolute percentage error) metric when the model is tested on the artificial dataset. The shaded area shows the 95% confidence interval.

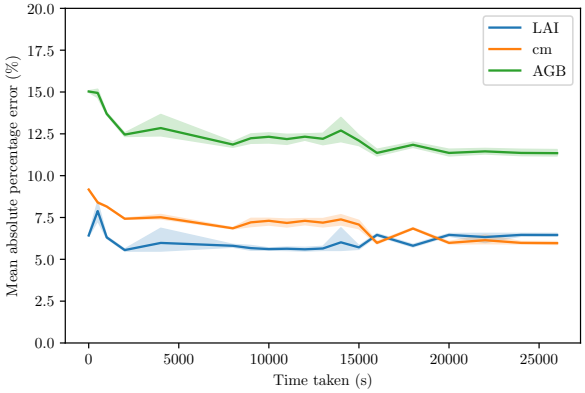


(a) Multi-output regressor

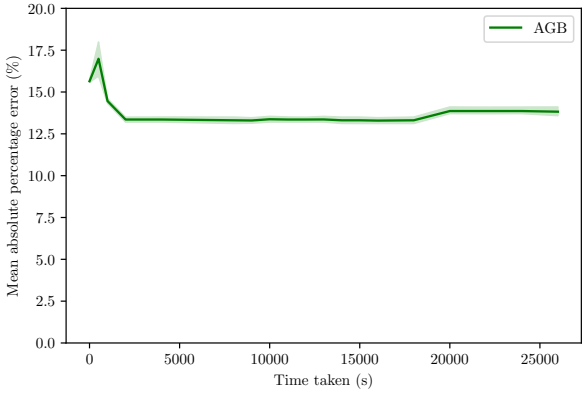


(b) Single-output regressor

Figure 18: Random forests

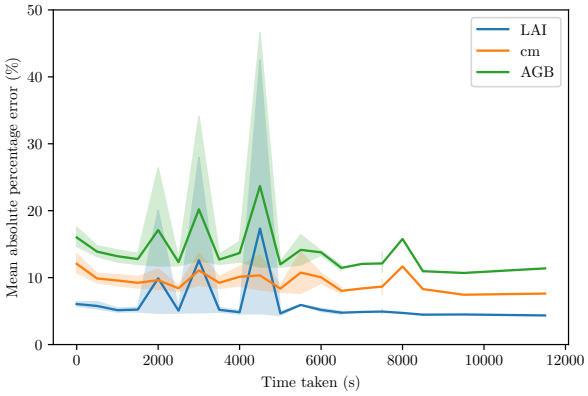


(a) Multi-output

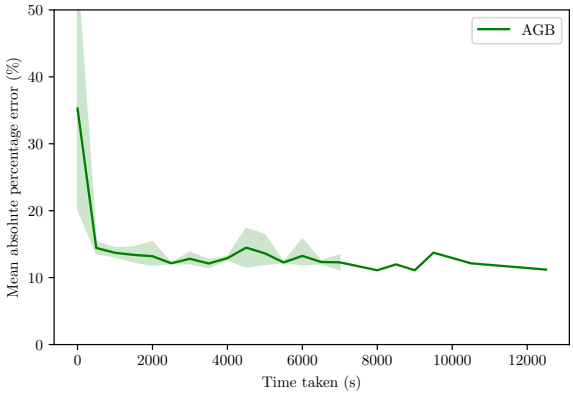


(b) Single-output

Figure 19: Gaussian process

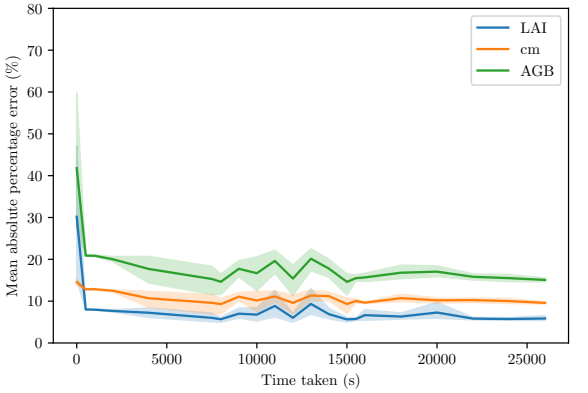


(a) Multi-output

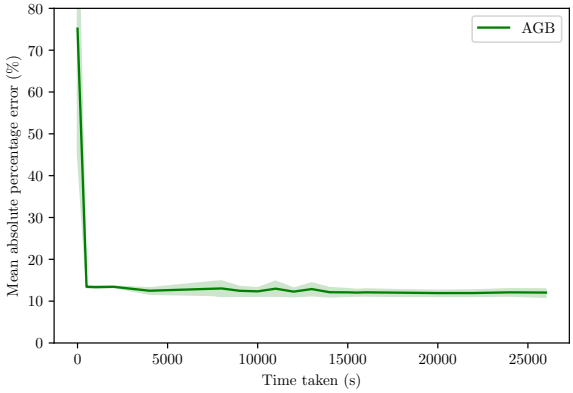


(b) Single-output

Figure 20: AutoKeras



(a) Multi-output



(b) Single-output

Figure 21: Auto-sklearn

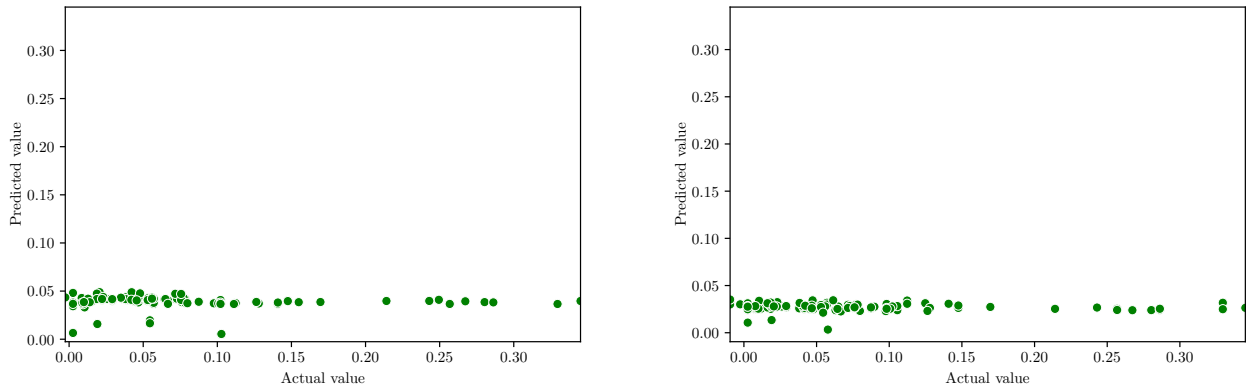
A.2 Field data

Table 12: The mean and standard deviation of the performance of the baseline models evaluated on the field dataset expressed in various metrics.

Model	AGB [g cm^{-2}] MAE	AGB [g cm^{-2}] RMSE	AGB [g cm^{-2}] MAPE
Random forests	0.0536 ($\sigma = 0.0006$)	0.0866 ($\sigma = 0.0008$)	205.9718 ($\sigma = 9.6346$)
Gaussian process	**0.0514 ($\sigma = 0.0002$)**	**0.0856 ($\sigma = 0.0002$)**	125.3716 ($\sigma = 1.4102$)
AutoKeras	0.0567 ($\sigma = 0.0116$)	0.0888 ($\sigma = 0.0125$)	141.4532 ($\sigma = 45.3825$)
Auto-sklearn	0.0543 ($\sigma = 0.0015$)	0.0878 ($\sigma = 0.0012$)	158.8600 ($\sigma = 14.0477$)
Random forests single-output	0.0532 ($\sigma = 0.0005$)	0.0860 ($\sigma = 0.0006$)	197.5492 ($\sigma = 12.2309$)
Gaussian process single-output	0.0516 ($\sigma = 0.0003$)	0.0860 ($\sigma = 0.0003$)	**119.0338 ($\sigma = 3.6652$)**
AutoKeras single-output	0.0558 ($\sigma = 0.0059$)	0.0897 ($\sigma = 0.0035$)	146.5579 ($\sigma = 143.9520$)
Auto-sklearn single-output	0.0539 ($\sigma = 0.0022$)	0.0880 ($\sigma = 0.0025$)	135.2766 ($\sigma = 23.2554$)

A.2.1 Scatterplots

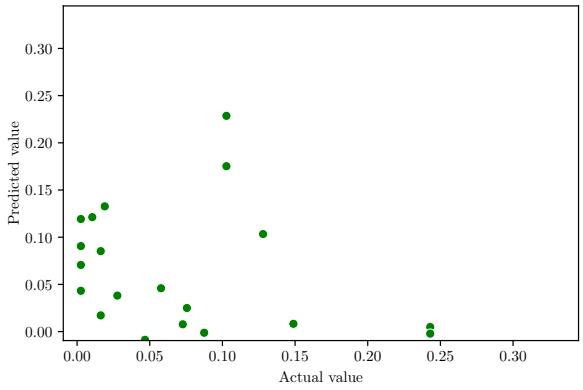
In this section scatterplots are presented hat show the performance for both the multi-output and single-output versions of all models. The models are evaluated on the field dataset and scatterplots are shown for every variable. All models were trained for the training time set as the baseline in Section 4. The x-axis relates to the actual value, while the y-axis relates to the estimated value.



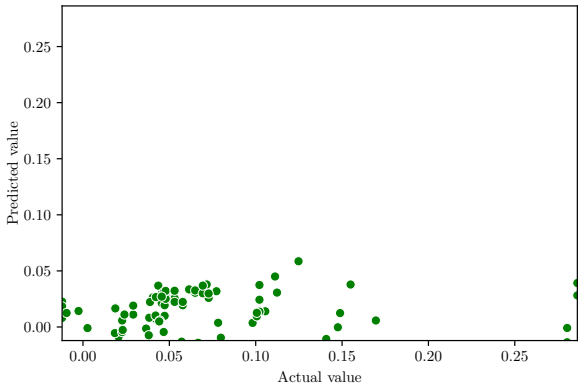
(a) AGB multi-output

(b) AGB single-output

Figure 22: Random forests

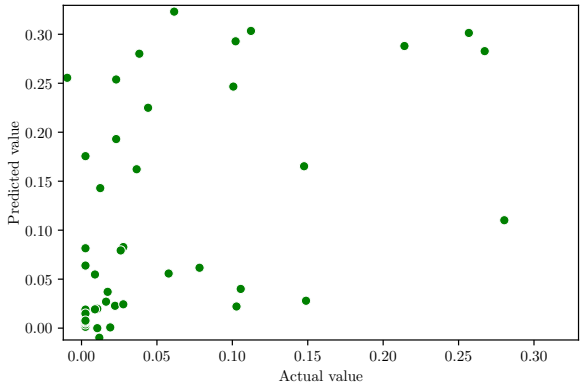


(a) AGB multi-output

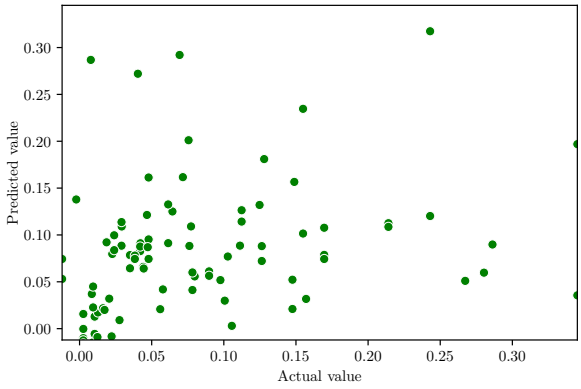


(b) AGB single-output

Figure 23: Gaussian process

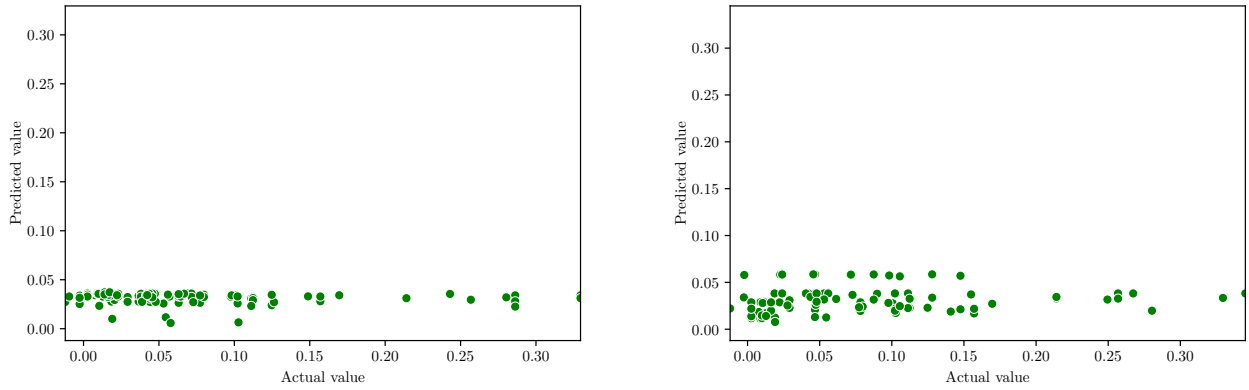


(a) AGB multi-output



(b) AGB single-output

Figure 24: AutoKeras



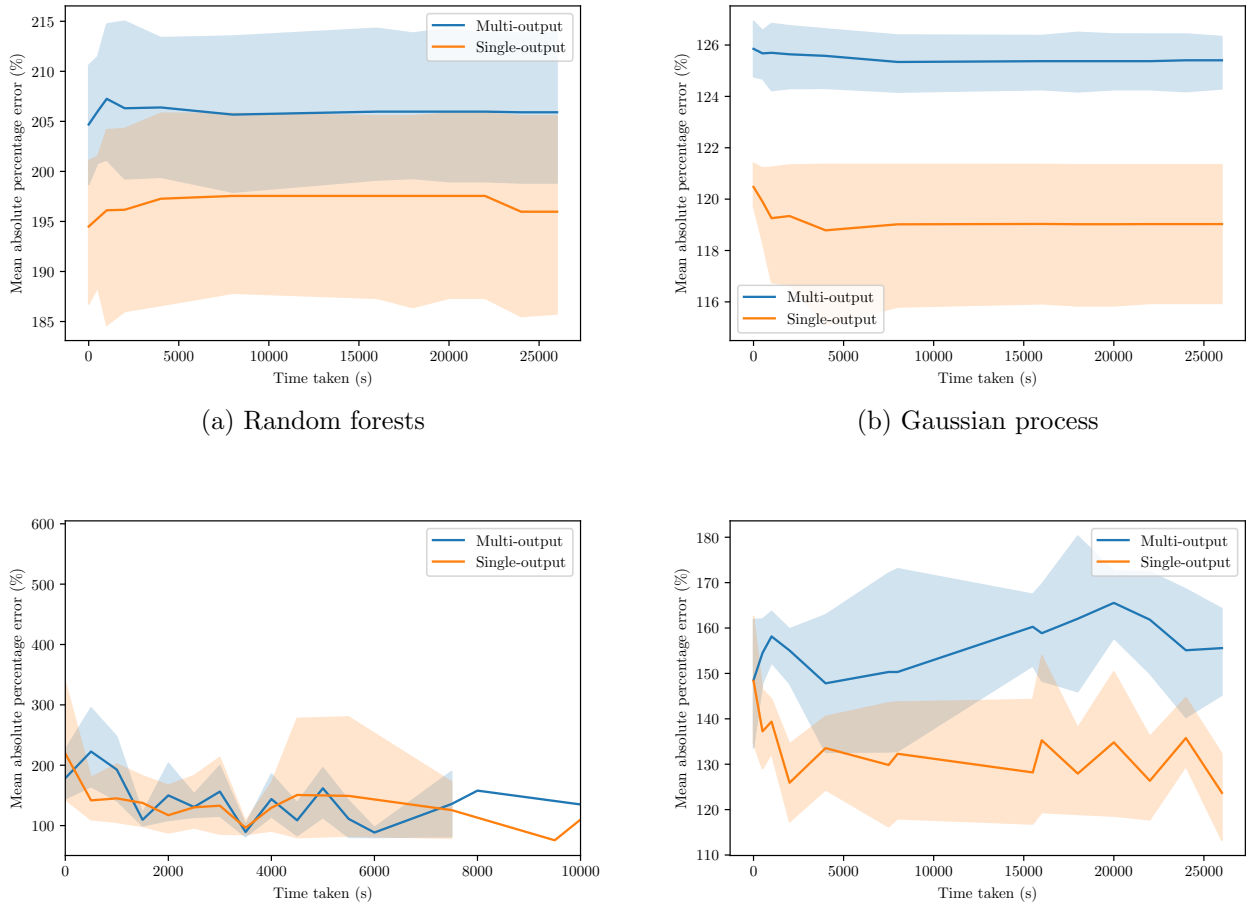
(a) AGB multi-output

(b) AGB single-output

Figure 25: Auto-sklearn

A.2.2 Relationship between training time and performance

For every model, the figures in this section show the relationship between the time taken for training the model and the performance of the model expressed using the MAPE (Mean absolute percentage error) metric when the model is tested on the field dataset. The shaded area shows the 95% confidence interval.



(a) Random forests

(b) Gaussian process

(c) AutoKeras

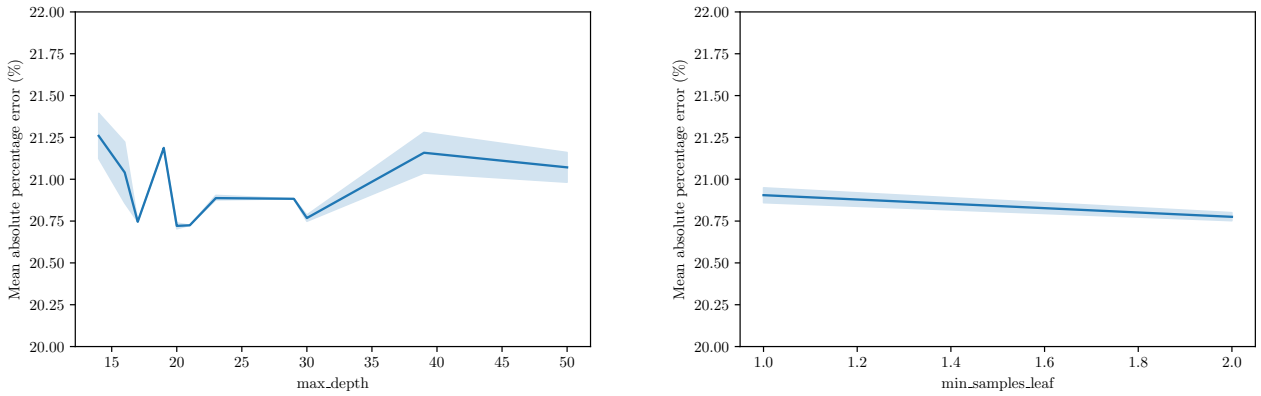
(d) Auto-sklearn

Figure 26: Training time compared to the Mean absolute percentage error (MAPE) performance of the output variables. The shaded area shows the 95% confidence interval.

A.3 Method-specific

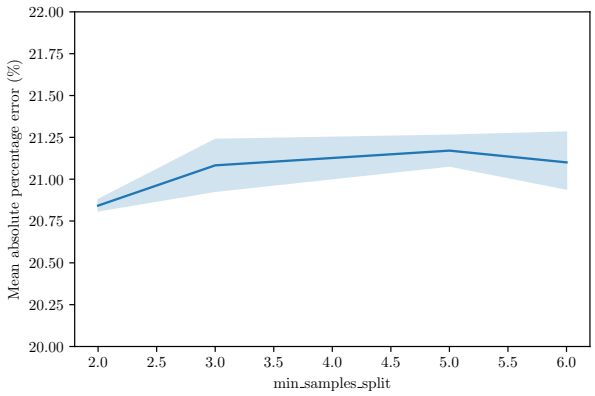
A.3.1 Random forests

In the following figure we show three line plots that show the impact of the three hyperparameters on the final MAPE for predicting the AGB value:



(a) `max_depth`

(b) `min_samples_leaf`

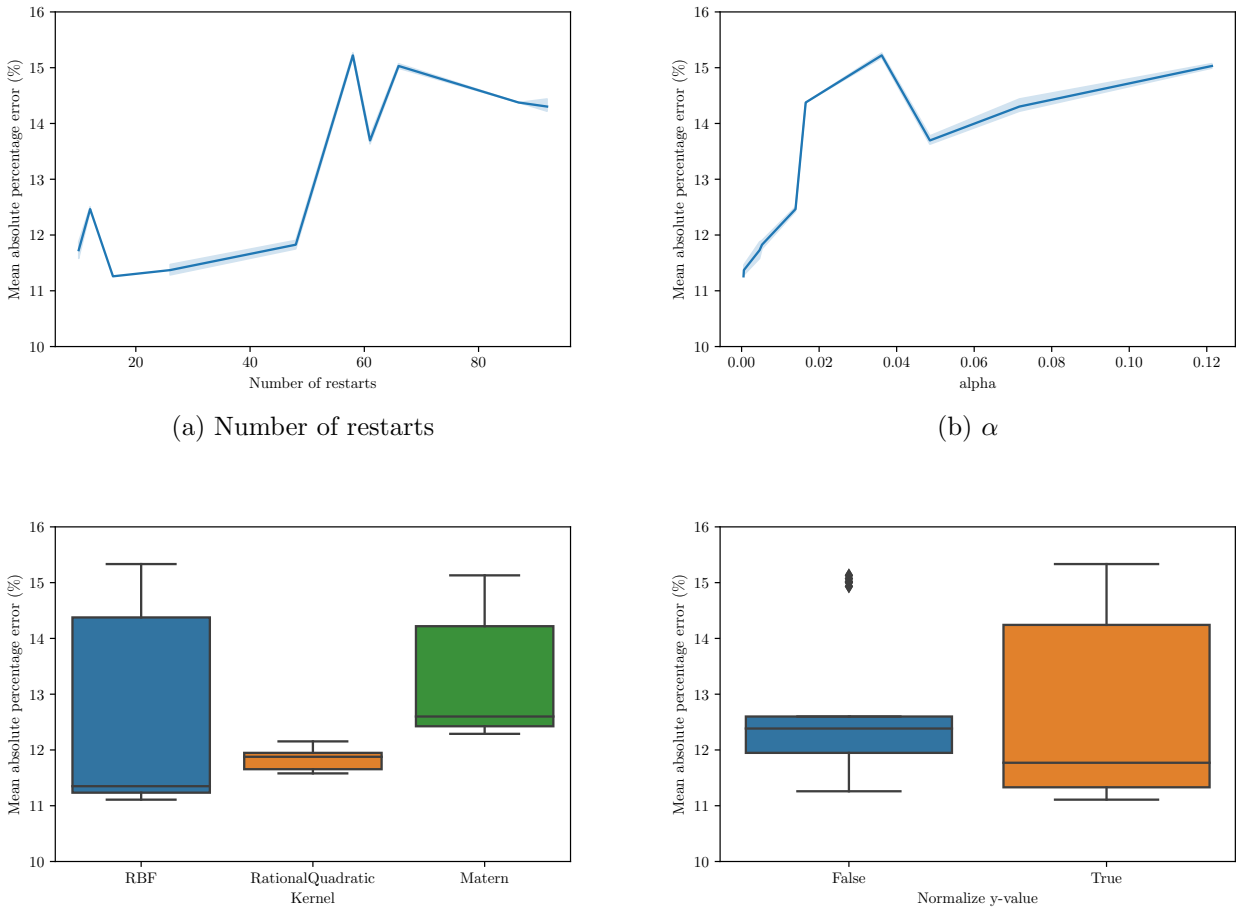


(c) `min_samples_split`

Figure 27: The impact of the hyperparameter that the random search optimises of the Random forests regressor.

A.3.2 Gaussian process

In the following figure we show four line plots that show the impact of the four hyperparameters on the final MAPE for predicting the AGB value:



(a) Number of restarts

(b) α

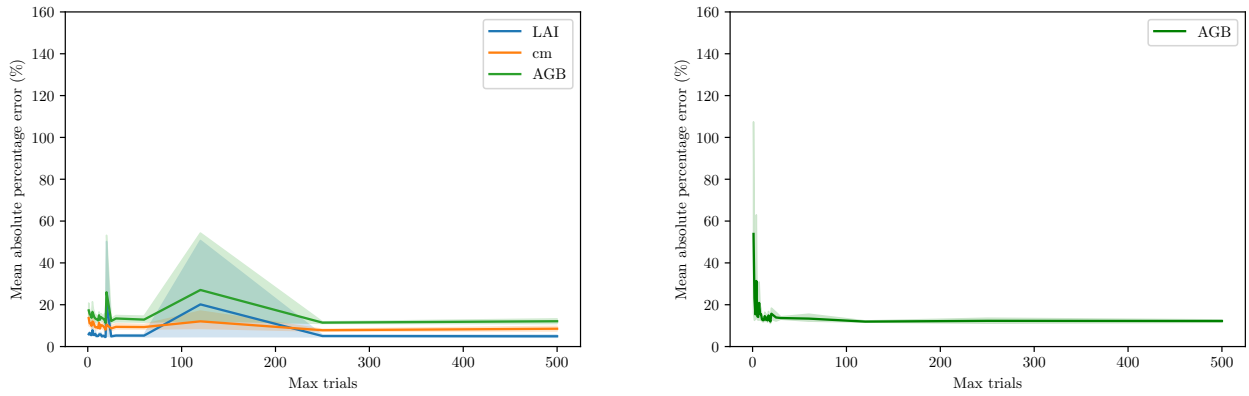
(c) Kernel

(d) Normalize y-value

Figure 28: The impact of the hyperparameter that the random search optimises of the Gaussian process regressor.

A.3.3 AutoKeras

AutoKeras was the only method that didn't provide a hyperparameter that directly controls the maximum training time. Instead, the training time can indirectly be controlled using the `max_trials` hyperparameter (see Section 3.3.2). In Figs. 29 and 30 we show similar figures to Figs. 20 and 26 respectively. However here we compare the value of the `max_trials` parameter against the MAPE, instead of the time taken for training against the MAPE. We see a much more stable and expected error curve, which decrease as the maximum amount of allowed trials increases.



(a) Multi-output

(b) Single-output

Figure 29: `max_trials` compared to the Mean absolute percentage error (MAPE) performance of the output variables.

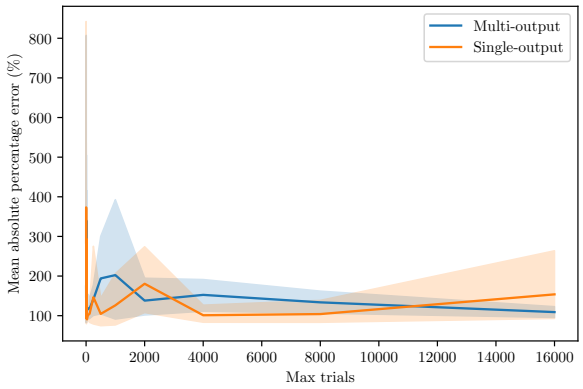
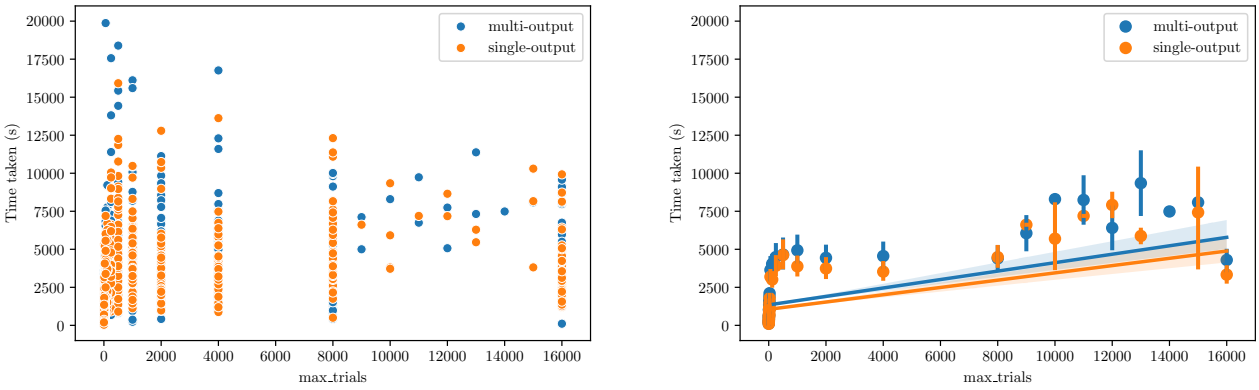


Figure 30: Mean absolute percentage error (MAPE) performance of the output variables compared against the value of the `max_trials` hyperparameter.

Since the `max_trials` parameter was used to indirectly control the maximum training time, it is useful to look at the relationship that exists between said parameter and the maximum training time. The following figure shows this relationship:



(a) Scatterplot

(b) Robust regression

Figure 31: The relationship between the `max_trials` hyperparameter and the time taken for training in the AutoKeras regressor.

Furthermore, the AutoKeras library automatically optimises the amount of epochs to used per trial. In the following figure it becomes obvious that there is no relationship between the `max_trials` hyperparameter and the amount of epochs, although one might expect one to exist:

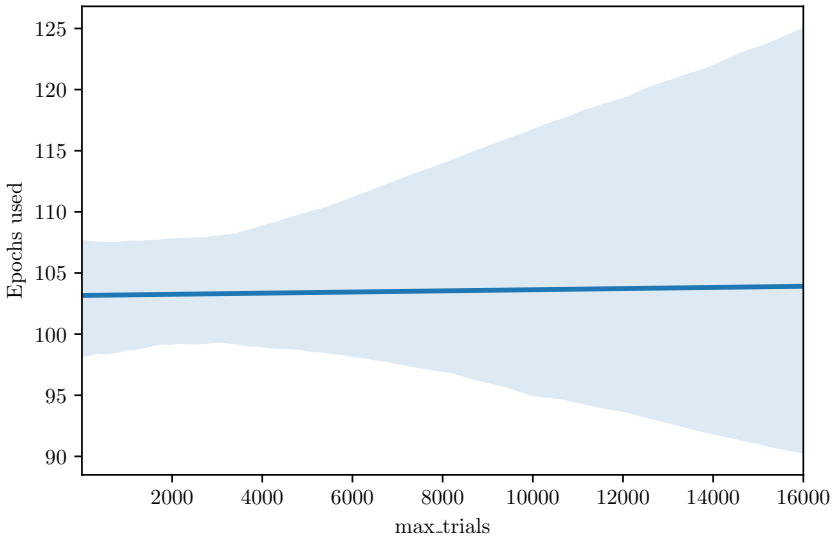


Figure 32: The relationship between the `max_trials` hyperparameter and the time taken for training in the AutoKeras regressor.

In Fig. 33 we show a Kernel Density Estimate (KDE) plot (combined with a rug plot) that shows the estimated density of the epochs as chosen automatically by AutoKeras. A peak can be seen around 100 epochs, with little skewness or kurtosis.

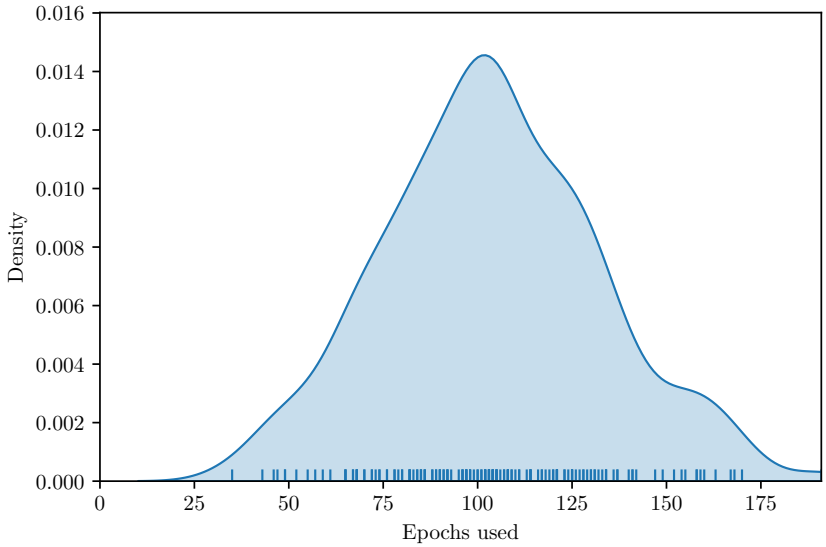


Figure 33: Kernel Density Estimate (KDE) of the epochs as chosen by AutoKeras. The rug plot shows the epochs used for every trained model.

B AdaNet

We originally also included the AdaNet [75] library in our comparison. However, it quickly became apparent that the library is too low level and thus out-of-scope for inclusion. Though the usage of AdaNet for this problem currently was too complicated, the library still shows potential in our opinion. In the GitHub repository, we also include a notebook containing novel code combining the AdaNet algorithm with Bayesian optimization (using the scikit-optimize library [29]) to find a Neural Network.

C Reproducibility

Care is taken to ensure the results are reproducible, all tools and notebooks that are required to produce the results that are shown in this thesis can be found at the authors' GitHub: `https://github.com/lieuwex/rtm-inversion-automl`.